



**FACULTAD DE POSTGRADO
TESIS DE POSTGRADO**

**EFICIENCIA EN LOS TIEMPOS DEL CICLO DE VIDA DEL
DESARROLLO DE SOFTWARE AL UTILIZAR ROLES EN EL
PERSONAL DE DESARROLLO DE FRONT-END**

SUSTENTADO POR:

Christian Alejandro Rodríguez Oseguera

PREVIA INVESTIDURA AL TÍTULO DE

GESTIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN

TEGUCIGALPA M.D.C., FRANCISCO MORAZÁN, HONDURAS,

C.A. Julio, 2022

UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

UNITEC

FACULTAD DE POSTGRADO

AUTORIDADES

UNIVERSITARIAS

RECTOR

MARLON ANTONIO BREVÉ REYES

VICERRECTOR ACADÉMICO NACIONAL

JAVIER ABRAHAM SALGADO LEZAMA

SECRETARIO GENERAL

ROGER MARTÍNEZ MIRALDA

DIRECTORA DE LA FACULTAD DE POSTGRADO

ANA DEL CARMEN RETTALLY VARGAS

**EFICIENCIA EN LOS TIEMPOS DEL CICLO DE VIDA DEL
DESARROLLO DE SOFTWARE AL UTILIZAR ROLES EN EL
PERSONAL DE DESARROLLO DE FRONT-END**

**TRABAJO PRESENTADO EN CUMPLIMIENTO DE LOS
REQUISITOS EXIGIDOS PARA OPTAR AL TÍTULO DE
MÁSTER EN GESTIÓN DE LAS TECNOLOGÍAS DE LA
INFORMACIÓN.**

ASESOR

JORGE RAÚL MARADIAGA CHIRINOS

MIEMBROS DE LA TERNA:

ROMÁN ARTURO PINEDA SOTO

RIGOBERTO RODRÍGUEZ



FACULTAD DE POSTGRADO

EFICIENCIA EN LOS TIEMPOS DEL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE AL UTILIZAR ROLES EN EL PERSONAL DE DESARROLLO DE FRONT-END

SUSTENTADO POR:

Christian Alejandro Rodríguez Oseguera

RESUMEN

El **propósito** de esta investigación pretende indagar y describir los posibles roles que se pueden desempeñar en un equipo de desarrollo de Front-End, buscando aumentar la eficiencia en los tiempos del SDLC. Se planteó como **objetivo** evaluar la posibilidad de involucrar al personal de Front-End en etapas tempranas del SDLC, iniciando desde el análisis del sistema y culminando en la etapa de implementación. Se empleó una **metodología** con enfoque cualitativo, alcance descriptivo y técnica de muestreo no probabilístico. El **análisis de resultados** describe las etapas del SDLC en las cuales el equipo de Front-End puede tener una participación y el rol o los roles que participan en dicha etapa.

Como **conclusión** principal, involucrar al personal de Front-End desde etapas tempranas del SLCD Aporta un valor agregado del servicio de tecnología, ayuda a la reducción de solicitudes de cambio y reduce los tiempos de desarrollo.

Palabras claves: (Front-End, software, maquetador, ITIL)



GRADUATE SCHOOL

**SOFTWARE DEVELOPMENT LIFECYCLE TIME EFFICIENCIES
BY UTILIZING ROLES IN THE FRONT-END DEVELOPMENT
WORKFORCE**

Author:

Christian Alejandro Rodríguez Oseguera

ABSTRACT

The **purpose** of this research is to investigate and describe the possible roles that can be played in a Front-End development team, seeking to increase efficiency in the SDLC. The **objective** was to evaluate the possibility of involving Front-End personnel in early stages of the SLDC, starting from the system analysis and culminating in the implementation stage. A **methodology** with a qualitative approach, descriptive scope and non-probabilistic. The **results analysis** describes the stages of the development life cycle in which the Front-End team can have an involvement and the role(s) involved in that stage.

As a main **conclusion**, involving Front-End personnel from early stages of the software development life cycle adds value to the technology service, helps to reduce change requests and reduces development times.

Keywords: Front-End, software, web designer, I

Índice General

CAPÍTULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN	1
1.1. INTRODUCCIÓN.....	1
1.2. DEFINICIÓN DEL PROBLEMA.....	3
1.2.1. <i>Antecedentes del problema</i>	3
1.2.2. <i>Planteamiento del problema</i>	4
1.2.3. <i>Preguntas de investigación</i>	5
1.3. OBJETIVOS DEL PROYECTO	6
1.3.1. <i>Objetivo general</i>	6
1.3.2. <i>Objetivos específicos</i>	6
1.4. JUSTIFICACIÓN.....	7
1.4.1. <i>Justificación teórica</i>	7
1.4.2. <i>Justificación practica</i>	7
CAPÍTULO II. MARCO TEÓRICO	8
2.1 CONCEPTUALIZACIÓN.....	8
2.2 ANÁLISIS DE LA SITUACIÓN ACTUAL.....	12
2.2.1. <i>Macro entorno</i>	12
2.2.2. <i>Micro entorno</i>	21
2.3. TEORÍAS DE SUSTENTO	23
2.3.1. <i>Análisis de las metodologías</i>	27
2.3.2. <i>Antecedentes de las metodologías</i>	29
2.3.3. <i>Análisis crítico de las metodologías</i>	30
2.4. HERRAMIENTAS	32
2.5. MARCO LEGAL	33
2.5.1. <i>Reglamento sobre el gobierno electrónico</i>	33
2.5.2. <i>Código penal</i>	36
CAPÍTULO III. METODOLOGÍAS DE LA INTESTIGACIÓN	39
3.1 TIPO DE ENFOQUE	39
3.2 ALCANCE	39
3.3 POBLACIÓN.....	40
3.4 MUESTRA.....	40
3.5 UNIDAD DE ANÁLISIS.....	41
3.5.1 <i>Criterios</i>	41
3.6 OPERACIONALIZACIÓN DE VARIABLES	42

3.7	INSTRUMENTOS	44
3.7.1	<i>Entrevistas semiestructuras</i>	44
3.7.2	<i>Consulta a expertos</i>	45
3.7.3	<i>Fuentes de información</i>	45
3.8	PLAN DE ANÁLISIS.....	46
3.9	MATRIZ METODOLÓGICA	47
CAPÍTULO VI. RESULTADOS Y ANÁLISIS		50
4.1	PROCESO ACTUAL.....	50
4.2	SDLC CON PARTICIPACIÓN INICIAL DE FRONT-END	54
4.3	RECOLECCIÓN DE DATOS.....	56
4.3.1	<i>Entrevistas</i>	56
4.3.2	<i>Consulta a expertos</i>	57
4.4	ANÁLISIS DE LOS RESULTADOS.....	57
4.4.1	<i>Proceso actual</i>	57
4.4.2	<i>SDLC con participación inicial de Front-End</i>	59
4.4.3	<i>Resultados según variables</i>	74
CAPÍTULO V. CONCLUSIONES Y RECOMENTACIONES		75
5.1	CONCLUSIONES.....	75
5.2	RECOMENDACIONES	78
CAPÍTULO VI. PROPUESTA		80
6.1	DEFINICIÓN Y USO DE ROLES EN EL EQUIPO DE FRONT-END	80
6.2	JUSTIFICACIÓN	80
6.3	ALCANCE	81
6.3.1	<i>Objetivo general</i>	81
6.3.2	<i>Objetivos específicos</i>	81
6.4	DESCRIPCIÓN Y DESARROLLO	82
6.4.1	<i>Descripción</i>	82
6.4.2	<i>Desarrollo</i>	84
6.5	MEDIDAS DE CONTROL.....	100
6.6	PROCESO DE IMPLEMENTACIÓN	102
BIBLIOGRAFÍA.....		104
ANEXO 1.....		107
	GUÍA PARA ENTREVISTA	107
ANEXO 2.....		108

GUÍA CONSULTA A EXPERTO.....	108
ANEXO 3.....	109
ENTREVISTA 1.....	109
ANEXO 4.....	113
GUÍA CONSULTA A EXPERTO.....	113

Índice de tablas

Tabla 1. Conceptos generales.	8
Tabla 2. Comparativa de herramientas de diseño	16
Tabla 3. Buenas prácticas de TI.....	18
Tabla 4. Operacionalización de variables	42
Tabla 5. Matriz metodológica	47
Tabla 6. Comparación de tiempos rol tradicional vs distintos roles de Front-End.....	71
Tabla 7. Condiciones sociales del diseñador	86
Tabla 8. Competencias Organizacionales	86
Tabla 9. Competencias Función/Técnicas	86
Tabla 10. Responsabilidades del diseñador	87
Tabla 11. Condiciones sociales del maquetaador	91
Tabla 12. Competencias Organizacionales del maquetaador	91
Tabla 13. Competencias Función/Técnicas del maquetaador.....	91
Tabla 14. Responsabilidades del maquetaador	92
Tabla 15. Condiciones sociales del programador lógico	95
Tabla 16. Competencias Organizacionales del programador lógico	95
Tabla 17. Competencias Función/Técnicas del programador lógico.....	95
Tabla 18. Responsabilidades del programador lógico	96

Índice de figuras

Figura 1. Estadísticas de actividad en internet.....	14
Figura 2. El informe sobre las tendencias digitales, redes sociales y mobile.	14
Figura 3. Popularidad de lenguajes de programación.....	17
Figura 4. Resumen revolución industrial.....	25
Figura 5. Ciclo de vida del desarrollo de software según ITIL.	26
Figura 6. Cinco fases de ITIL.	31
Figura 7. Propuesta de “coreografía” del análisis cualitativo.....	46
Figura 8. Herramientas de diseño de pantallas.	51
Figura 9. Diseño de pantalla en Pencil.....	52
Figura 10. Participación de Front-End en el SDLC.	55
Figura 11. Diseño de pantalla en Figma.....	56
Figura 12. Componentes que afectan la eficiencia del SDLC.	58
Figura 13. Flujo de actividades del diseñador.	62
Figura 14. Etapas del SDLC en las que el diseñador participa.	63
Figura 15. Flujo de actividades del maquetador.	65
Figura 16. Etapas del SDLC en las que el maquetador participa.....	66
Figura 17. Flujo de actividades de el programador lógico.....	68
Figura 18. Etapas del SDLC en las que el programador lógico participa.....	69
Figura 19. Grid con funciones básicas.....	72
Figura 20. Botones y diseño con hover.....	72
Figura 21. Drop Down List con funcionalidad básica.....	73
Figura 22. Input con espacio para un icono.	73
Figura 23. Modal vacío con botones inferiores.	73
Figura 24: Proceso de implementación de roles en el Front-End.....	103

CAPÍTULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1. Introducción

La presente investigación se refiere a la participación del equipo de Front-End en las distintas etapas del ciclo de vida del desarrollo de software, en la cual describimos como el componente de **Front-End** a evolucionado desde desarrollar páginas web informativas a desarrollar sistemas o aplicaciones web, en esta evolución han surgido nuevas tecnologías, metodologías y actividades que han sido analizadas con el objetivo de eficientes los tiempos de desarrollo en el SDLC.

Para analizar esta problemática es necesario mencionar sus causas. Una de ellas es el aumento en las solicitudes de cambio por parte de los usuarios de negocio, se entiende por solicitud de cambio a todas las peticiones formales por parte de los usuarios de negocio para crear, modificar o eliminar pantallas o funcionalidades de un sistema. Estas solicitudes de cambio se están realizando en tiempos cortos posteriores a la aprobación de un sistema, otra de las causas es la gran cantidad de tiempo requerido para completar un ciclo de vida de desarrollo de software en el BCH.

El **interés al nivel profesional**, como líder técnico o gestor de proyectos, el interés es poder reducir los tiempos de desarrollo y en todo el ciclo de vida de desarrollo de software es muy importante, implementado roles de formenterano los cuales participen en las distintas etapas del SDLC ayudará a aumentar la calidad, la satisfacción en los usuarios de negocio y eficientes los tiempos en el SDLC.

La **metodología de la investigación** se utilizó un enfoque predominante cualitativo para

describir los roles a implementar en el equipo de Front-End los cuales tienen participación en las distintas etapas del SDLC, Con un alcance descriptivo el cual se emplea sobre una muestra no probabilística, el cual se realizó en la sección de proyectos de banca central a los analistas y gestores de proyecto, utilizando entrevistas semi-estructuradas y consulta a expertos para la recolección de datos.

Cómo **finalidad para realizar este trabajo de investigación** está el poder evaluar en el actual ciclo de vida del desarrollo de software del área de desarrollo del BCH, Describiendo las etapas del SLC en las cuales participará el equipo de Front-End y poder definir las actividades y funciones que tendrán cada uno de estos roles. De esta forma crear los **descriptores de puestos y perfiles** que documenten los roles y sus funciones.

La estructura de este documento está compuesta por el **capítulo uno** el cual nos muestra la formulación del problema y su objetivo de investigación. El **capítulo dos** nos brinda conceptos y teorías sobre el ciclo de vida del desarrollo de software, ITIL versión cuatro, Front End y sus herramientas. El **capítulo tres** nos detalla la metodología de investigación empleada. El **capítulo cuatro** recopila todos los datos, los analiza y los describe haciendo uso de gráficos y tablas. El **capítulo cinco** presenta las conclusiones y recomendaciones. Finalmente, el **capítulo seis** presenta una propuesta de implementación.

1.2. Definición del problema

1.2.1. Antecedentes del problema

La ingeniería web está en continua evolución y un ejemplo de ello fue la implementación de aplicaciones web, estas implementaciones se han logrado desacoplando los componentes del software a un servidor de aplicación (Back-End) y un navegador de internet (Front-End). Esto con el fin de estructurar mejor las aplicaciones web, para un mejor rendimiento y estructura, para ello existen una gran variedad de tecnologías que ayudan el desarrollo de los componentes previamente mencionados. Actualmente nos limitamos a definir el componente de Front-End sólo como el diseño que el usuario final observa, pero hay muchas funciones detrás de ello, funciones que pueden implementarse desde la redacción de los requerimientos de usuario, para definir visualmente las salidas y el producto final del sistema, hasta una mejor visualización de la necesidad de negocio que se desea satisfacer. Esto ayudará a no obtener un gran número de solicitudes de cambio, o lograr las expectativas del cliente al entregar el sistema.

El Banco Central cuenta con metodologías y estándares internos a seguir para el desarrollo de sus sistemas, pero como la implementación de esta nueva arquitectura es reciente no hay funciones definidas para los desarrolladores de Front-End lo cual tiene un impacto en el ciclo de vida del desarrollo de software (SDLC), creando incertidumbre cuando se definen los rangos de tiempos para realizar algunas actividades.

1.2.2. Planteamiento del problema

En la implementación de una nueva arquitectura en la cual se contempla Back-End y Front- End como componentes separados, se ha identificado que estos componentes aun no pueden trabajar de forma simultánea e independiente uno del otro, consecuencia de esto en algunas ocasiones se puede gestionar de una manera errónea el tiempo de desarrollo, lo cual incurre en mayores costos finales del producto y retrasos. Dichos inconvenientes se producen ya que no existen roles ni sus funciones bien definidas para los desarrolladores del equipo de Front-End, Asimismo, no existen métricas que puedan ayudar a la evaluación de la capacidad y el control de estos equipos.

Debido a la situación problemática presentada, existe la necesidad de definir los roles y las funciones que tendrán el equipo de desarrollo de Front-End, para que sean compatibles con el ciclo de vida del desarrollo de un sistema. Para dicho fin se deberá definir tres roles principales que ayuden en todas las etapas del ciclo de vida del desarrollo de un sistema.

Por lo tanto, en relación con los roles del equipo de desarrollo de Front-End se plantea la siguiente pregunta:

¿Involucrar al equipo de Front-End desde el inicio del ciclo de vida de desarrollo del software del BCH, mejorará a eficiencia en los tiempos de entrega del software y la satisfacción del cliente?
--

1.2.3. Preguntas de investigación

1. ¿El ciclo de vida de desarrollo de un sistema utilizado por el equipo de Front-End es el adecuado para poder acoplar o integrar los cambios que se proponen?
2. ¿La implementación de roles en el equipo de desarrollo de Front-End del BCH, tendrá el impacto suficiente en cuanto a tiempos de entrega a la hora de afrontar un desarrollo?
3. ¿Dada la situación actual del equipo de desarrollo de Front-End, podría de cierta forma ser contraproducente el alterar como ellos están acostumbrados a realizar su trabajo, o como estos cambios se integran con los demás equipos que tienen relación con este?

1.3. Objetivos del proyecto

1.3.1. Objetivo general

Evaluar el ciclo de vida del desarrollo de software del área de desarrollo del BCH, para mejorar la eficiencia en los tiempos, utilizando roles en el personal de desarrollo de Front-End.

1.3.2. Objetivos específicos

1. Identificar que componentes pueden estar afectando la eficiencia, ciclo de vida del desarrollo de software del área de desarrollo del BCH.
2. Describir como el uso de roles en el equipo de Front-End puede reducir los tiempos de desarrollo en los proyectos.
3. Definir roles y sus funciones para el equipo de desarrollo del Front-End del BCH.
4. Analizar e identificar las etapas del ciclo de vida del desarrollo de software en las cuales la participación del equipo de desarrollo del Front-End produzca una mejoría.

1.4. Justificación

1.4.1. Justificación teórica

La presente investigación se realizó con la finalidad de demostrar como mediante el análisis del ciclo de vida del desarrollo de software que utiliza el BCH e integración del equipo de desarrollo de Front-End pueden tener un alto impacto en tiempos de entrega de los proyectos, mejorar eficiencia del equipo al igual que crear una mejor experiencia para el usuario. Asimismo, se definieron actividades y responsabilidades para cada rol de los integrantes del equipo de desarrollo de Front-End.

1.4.2. Justificación practica

Nuestra investigación fue realizada debido a que se detectó la necesidad de establecer un mayor control dentro del equipo de desarrollo de Front-End el cual permita a todo el equipo poder centrarse en sus actividades correspondientes lo cual genera mayor agilidad, eficiencia y mejora en los tiempos. Esto permitió poder identificar actividades en las cuales el equipo de Front-End podría tener una mayor participación. Partiendo del análisis del ciclo de vida de desarrollo continuando con la implementación de roles los cuales serán el pilar de nuestra investigación, estos fueron elaborados partiendo del uso de mejores prácticas, estándares internacionales y consulta a expertos.

CAPÍTULO II. MARCO TEÓRICO

2.1 Conceptualización

Tabla 1. Conceptos generales.

Concepto	Definición
Agile	Enfoque iterativo que, para el desarrollo de software, enfocándose en pequeñas entregas en el menos tiempo posible.
AHIBA	Asociación Hondureña de instituciones bancarias.
API	Interfaz de Programa de Aplicación
Azure	Servicio de cómputo creado por Microsoft con el fin de ayudar a la construcción, prueba despliegue y administración de aplicaciones.
Back-End	Componente de un sistema que se encarga de gestionar y estructurar la información mostrada por el componente de Front-End. Este componente no es accesible por los usuarios finales.
BANHPROVI	Banco Hondureño para la Producción y vivienda
BCH	Banco Central de Honduras
Branch - Rama	Versión de código compuesta por uno o varios commits, esta rama permite segmentar el repositorio y con ello lograr un repositorio descentralizado.
CD	Despliegue Continuo, en termino de desarrollo de software equivale a compilar y desplegar una aplicación de manera automática.
CI	Integración Continua, en términos de desarrollo de software equivale a fusionar o combinar distintas versiones de código de manera

	automática cada vez que sea necesario.
CNBS	Comisión Nacional de Bancos y Seguros.
Commmmit	Termino empleado en el desarrollo de sistemas, para confirmar que un cambio se almacene en un repositorio central o ejecutar un comando en base de datos.
Core Banking	Es el sistema informático principal utilizado por las instituciones financieras.
DevOps	Conjunto de prácticas que agrupas el desarrollo de sistemas y las operaciones de TI, con el objetivo de eficientar el ciclo de vida del desarrollo de software (SDCL).
Docker	Aplicación que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.
Docker Hub	Repositorio de imágenes Docker publicas el cual está alojado en la nube y es proporcionado por el proveedor de Docker.
DockerFile	Archivo que contiene las instrucciones o pasos a seguir, para la creación de imágenes Docker. Este archivo usualmente está ubicado en el mismo repositorio del código fuente de la aplicación.
DockerRegistry	Repositorio de imágenes Docker privadas, en cual los usuarios pueden almacenar o descargar imágenes Docker.
Fetch	Comando que descarga commits, archivos y referencias de un repositorio remoto a un repositorio local.

Framework	Es un software que ofrece una más o funcionalidades específicas que se pueden utilizar como base o ayuda para desarrollar sistemas de información o interfaces.
Front-End	Componente de un sistema que se encarga de mostrar la información de manera atractiva y simple al usuario final.
GIT	Software que ayuda a controlar el versionamiento e historial de un conjunto de archivos o código fuente.
Hardware	Son los componentes físicos de un equipo de cómputo.
HTML	HyperText Markup Language.
Interbanca	Nombre que se le designa a la banca por internet, cuya principal herramienta es una aplicación web la cual contiene los productos y servicios del banco.
Interfaz	Conjunto de elementos que permiten visualizar de manera grafica las funcionalidades de un sistema.
ITIL	Biblioteca de Infraestructura de Tecnologías de Información.
Kernel	Es un software que constituye una parte fundamental del sistema operativo, y se define como la parte que se ejecuta en modo privilegiado.
Kubernetes	Sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores que fue originalmente diseñado por Google y donado a la Cloud Native Computing Foundation.
Maquetado	Proceso en el cual se replicada la apariencia de un diseño, en HTML y

	CSS para convertirlo en la interfaz de un sistema.
Merge	Unión de dos versiones o ramas de código fuente y así crear u obtener una nueva versión de código fuente.
En-Premise	Termino empleado para describir ubicación física de un centro de cómputo o infraestructura tecnológica.
Pipeline	Conjunto de tareas que se realizan de manera automática o manual, las cuales están relacionadas con el despliegue continuo (CD) de una aplicación web.
Prototipo	Diseño dinámico de una pantalla, el cual muestra el posible flujo que tendrá un sistema.
Pull	Comando que se emplea para extraer y descargar contenido desde un repositorio remoto y actualizar al instante el repositorio local para reflejar ese contenido.
Push	Comando que se usa para cargar contenido del repositorio local a un repositorio remoto.
Realese	Despliegue de una aplicación
SDLC	Ciclo de vida del desarrollo de software (System Development Life Cycle)
Software	Comprende todo el conjunto de programas, procedimientos y rutinas asociados al funcionamiento de un sistema informático.
Staging	Rama previa a la rama de producción.
SVS	Sistema de valor del servicio. utilizado en ITIL versión cuatro
TFVC	Team Foundation Version Control

UNITEC	Universidad Tecnológica Centroamericana
URL	Localizador de recurso uniforme (Uniform Resource Locator), el cual se utiliza para la navegación en distintas páginas web en internet.
Webinars	Conferencia o seminario de manera online, este puede ser grabado o en vivo.

2.2 Análisis de la situación actual

2.2.1. Macro entorno

El desarrollo de aplicaciones web ha tenido una gran demanda a nivel mundial, por lo que han surgido nuevas herramientas, estándares, puestos, arquitecturas para hacer el desarrollo de aplicaciones más eficiente y con mayor calidad, para ello plantearémos un análisis de la situación actual a nivel mundial, describiendo distintos factores que influyen en el desarrollo de las aplicaciones web.

2.2.1.1 Front-End

Front-End es el componente que se encarga principalmente de la apariencia de un sistema web o de escritorio, aunque sus funciones no solo se limitan a la apariencia o diseño de un sistema, ya que también analizan la forma en la que el usuario final va a interactuar con la aplicación, como el usuario ingresará los datos y cual será la estructura que tendrán los datos previos a almacenarlos en una base de datos y de esta manera poder comunicarse exitosamente con el Back-End por medio de API's.

Front-End se enfoca en dos ámbitos como ser: UX y UD, el primero UX es la experiencia que espera y/o recibe el usuario al utilizar la aplicación, para ello Front-End tendrá contacto con el usuario antes, durante y hasta después de desarrollar las aplicaciones para comprobar que el sistema cumpla con sus expectativas y la segunda UD es el diseño o

apariciencia que tendrá la aplicación, plasmándola primero en una imagen y luego programar una interfaz en HTML, CSS y JS.

2.2.2.2 Aplicaciones Web

El desarrollo de aplicaciones a nivel internacional es un área en continua evolución, en la cual se utilizan distintas tecnologías, arquitecturas y herramientas con el fin de mejorar el desarrollo, rendimiento y apariciencia de aplicaciones web. Al enfocarnos en el componente de Front-End existen conferencias en distintas regiones del mundo como ser, Europa, Norteamérica, Suramérica y Asia, algunas de las conferencias más relevantes que se realizan anualmente son:

- SpainJS (Madrid, España)
- ConveyUX (Seattle, USA)
- Cumbre de Adobe (Las Vegas, USA)
- Google Cloud Next (San Francisco, USA)
- Codemotion (Online)
- Festival La próxima web de Europa (Ámsterdam)

Estas conferencias reúnen expertos en distintas áreas de conocimiento relacionadas a Front-End, que ayudan a que los analistas y desarrolladores de este componente tengan un panorama mucho más completo en las tecnologías actuales y no que no se limiten a diseñar y maquetar las interfaces de una aplicaciones web, estos conocimientos son muy diversos como ser: diseño de interfaces, inteligencia artificial, análisis de las experiencias de usuarios, transparencias de pantallas, Framework en tendencia, robots, gestión de proyectos, talleres funcionales, gestión en el ciclo de vida del desarrollo de software, implicaciones en la administración del tiempo, entre otros.

Actualmente existen 1,927,521,572 sitios webs en internet, y con un incremento diario aproximado de 250,000 sitios web, por lo que demanda a los desarrolladores a estar en constante capacitación e investigación, para poder mantener sus aplicaciones actualizadas.



Figura 1. Estadísticas de actividad en internet

Fuente: <https://www.internetlivestats.com>

Producto a la pandemia del COVID-19 la cual se originó a inicio del año 2019, se ha visto un incremento relevante en la actividad de usuarios activos en redes sociales, sitios y aplicativos webs, como resultado las empresas buscan estar presente en estas plataformas y esto incurre en mayor demanda de desarrolladores web.

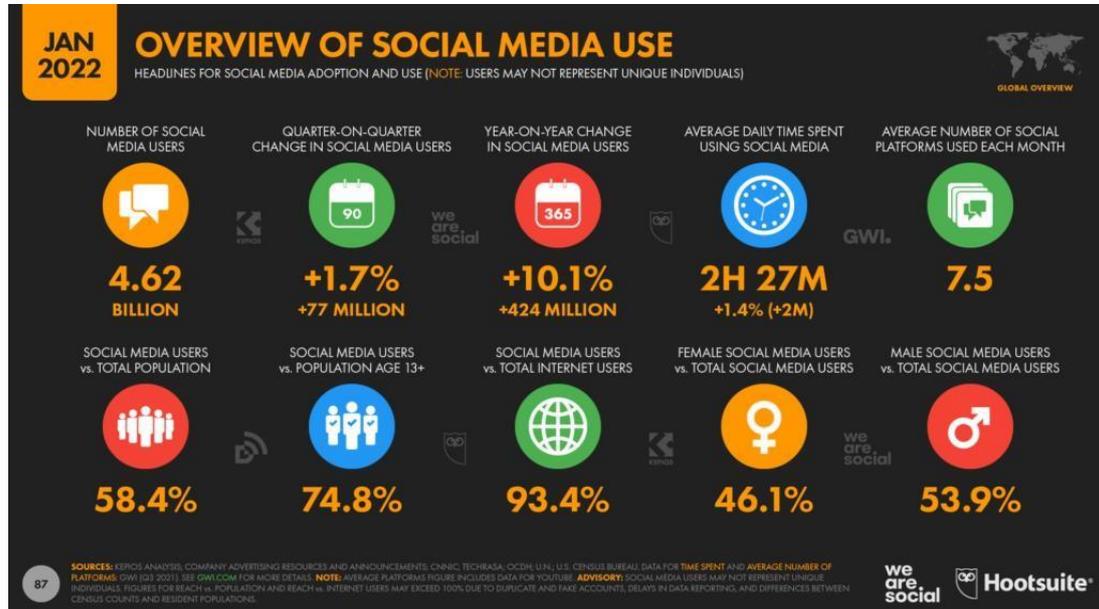


Figura 2. El informe sobre las tendencias digitales, redes sociales y mobile.

Fuente: Digital 2022 de Hootsuite y We Are Social

2.2.2.3 Framework

Al analizar el creciente mercado de aplicaciones y sitios web, se vio la necesidad de utilización de un Framework que facilite y agilice el desarrollo de estos aplicativos. En la actualidad existen diversos Framework en el mercado entre los cuales los más populares son:

1. **Angular:** Es un software de código abierto creado y respaldado por Google, este Framework dirigido a aplicación de una sola página o mejor conocidas como SPA por sus siglas en inglés (Single Page Application). Uno de los principios de este Framework se basa en estar compuesto de pequeños componentes reutilizables en toda la aplicación, los cuales permitan la construcción de aplicaciones livianas y escalables horizontalmente.
2. **React:** Es un software de código abierto creado y respaldado por Facebook. Esta librería está diseñada para construir interfaces de usuario dinámicas y que puedan detectar cambios en tiempo real sin la necesidad de actualizar componentes innecesarios o toda la interfaz.
3. **Vue:** Es un software de código abierto con licenciamiento MIT, el cual se creó con el fin de simular, pero de manera simplificada el funcionamiento de Angular. Este Framework se desarrolló de manera desacoplada con el fin de agregar únicamente las funcionalidades necesarias.

2.2.1.2 Herramientas

Implementar herramientas que ayuden a agilizar el ciclo de vida de desarrollo de software, es fundamental para crear aplicaciones y sitios web de calidad. Hacer uso de herramientas que agilicen el desarrollo, creación requerimientos y la administración de un proyecto, provocara involucrar a desarrolladores, sponsor, gestores, usuarios y gerentes de proyecto con el fin de crear aplicativos webs de mayor calidad.

Tabla 2. Comparativa de herramientas de diseño

Característica	Adobe XD	FIGMA	SKETCH
Versión gratis	Trial	SI	Trial
Precio individual	\$9.99	\$15	\$9
Precio grupal	\$22.99	\$45	\$9
SO	Linux, Mac y Windows	Linux, Mac y Windows	Mac
Plugin	SI	SI	SI
Colaboración	SI	SI	SI
Offline	SI	Limitado	SI
Prototipado	SI	SI	SI
UX	NO	SI	NO

Principales lenguajes de programación utilizados al 2022 en comparación al 2021, en el cual podemos observar que el lenguaje de Python ha tenido un incremento en el rating, ya que es utilizado para muchos sistemas estadísticos o de análisis de big data.

Feb 2022	Feb 2021	Change	Programming Language	Ratings	Change
1	3	↑	 Python	15.33%	+4.47%
2	1	↓	 C	14.08%	-2.26%
3	2	↓	 Java	12.13%	+0.84%
4	4		 C++	8.01%	+1.13%
5	5		 C#	5.37%	+0.93%
6	6		 Visual Basic	5.23%	+0.90%
7	7		 JavaScript	1.83%	-0.45%
8	8		 PHP	1.79%	+0.04%
9	10	↑	 Assembly language	1.60%	-0.06%
10	9	↓	 SQL	1.55%	-0.18%
11	13	↑	 Go	1.23%	-0.05%
12	15	↑	 Swift	1.18%	+0.04%
13	11	↓	 R	1.11%	-0.45%
14	16	↑	 MATLAB	1.03%	-0.03%
15	17	↑	 Delphi/Object Pascal	0.90%	-0.12%
16	14	↓	 Ruby	0.89%	-0.35%
17	18	↑	 Classic Visual Basic	0.83%	-0.18%
18	20	↑	 Objective-C	0.81%	-0.08%
19	19		 Perl	0.79%	-0.13%
20	12	↓	 Groovy	0.74%	-0.76%

Figura 3. Popularidad de lenguajes de programación
Fuente: TIOBE - The Software Quality Company, 2022

2.2.2.4 Buenas prácticas

Al crear sistemas web que son de uso internacional, es necesario implementar buenas prácticas al desarrollar dichos sistemas, para ello existen estándares y buenas prácticas a nivel mundial, entre ellas se encuentran las ISO 27000, COBIT, ITIL, BS 25999, entre otras.

Tabla 3. Buenas prácticas de TI

Estándar	Definición
ISO 27001	Es un estándar para la seguridad de la información denominado ISO/IEC 27001
ISO 27002	Se trata de una guía de buenas prácticas a partir de objetivos de control y controles recomendables a nivel de seguridad de la información.
ISO 27005	Se trata de un estándar internacional denominado ISO 27005:2008, Information technology – Security Techniques – Information Security Risk Management.
COBIT	Establece un marco de trabajo basado en dominios y procesos, a través del cual se ofrecen unas buenas prácticas enfocadas a optimizar la inversión de recursos en áreas de IT.
ITIL	Se aborda recursos orientados a la correcta gestión de los servicios de IT a través de un ciclo de vida de los servicios
BS 25999	Aborda los lineamientos que deben contemplarse para la administración de la continuidad del negocio

2.2.2.5 Estándares

- Microsoft Solución Framework (MSF) 4.0
- Norma para la Gestión y Control de Cambios en Tecnología de Información (TI)
- Proceso para la Gestión y Control de Cambios en Tecnología de Información (TI).
- Política para la Adquisición de Soluciones de Clase Mundial.
- Proceso para la Adquisición de Soluciones de Clase Mundial.
- Guía del PMBOK – Fundamentos de Gestión de proyectos (Project Management Body of Knowledge, por sus siglas en inglés) quinta edición

Producto que el desarrollo de Front-End cada día se está optimizando más, y las responsabilidades que este componente tiene son aún mayores, entre ellas el rendimiento una aplicación web, la apariencia, la experiencia del usuario, atraer al usuario, manipular los datos, entre otras, para ello el desarrollo de Front-End es realizado por un equipo en el cual se implementan algunos roles para eficientar los tiempos de desarrollo, entre los roles principales se encuentran los tres siguientes:

- Diseñador: Es la persona encargada de crear en imágenes la posible apariencia que tendrá la aplicación Web al finalizar el desarrollo.
- Maquetador: Es la persona encargada de recrear y construir en HTML y CSS los diseños o prototipos entregados por el personal del rol de diseñador y que estos fuesen aprobados por el usuario.
- Programador: Es el responsable de aplicar la lógica y estructurar los datos para la visualización de los mismos, también es el responsable de obtener y entregar la información correcta a una API.

2.2.2.6 Ciclo de vida de desarrollo de software

El ciclo de vida del desarrollo de software está compuesto por distintas etapas que van desde la socialización o solicitud de un sistema hasta la entrega de este al usuario. Estas etapas involucran a los usuarios que utilizarán el sistema, pero también a la parte administrativa de esa sección, asimismo, puestos ejecutivos del departamento de tecnología como a los desarrolladores que diseñarán y codificarán el sistema.

El ciclo de vida del desarrollo del software está compuesto por 10 etapas tradicionalmente las cuales pueden ajustarse o crear nuevas según la necesidad de cada empresa entre ellas destacan: solicitud del sistema, estudio de viabilidad, análisis del sistema, planificación, diseño, desarrollo, pruebas, implementación y mantenimiento.

- Solicitud del sistema: en esta etapa el usuario o cliente solicita o informa la necesidad de un nuevo sistema a el departamento de tecnología.
- Estudio de viabilidad: luego de recibir la solicitud se tiene comunicación con el cliente para analizar los requerimientos del sistema a grandes rasgos y poder analizar el impacto que tendrá el mismo.
- Análisis del sistema: en este paso se asignan recursos humanos de parte del cliente y de parte del departamento de tecnología para buscar y estudiar los requerimientos del sistema y de esta manera identificar limitaciones e impactos que tendrá sobre toda la organización el proyecto.
- Planificación: en esta etapa se planifica las actividades reuniones posibles entregas, investigaciones, prácticas y demostraciones que se realizarán en todo el ciclo de vida del desarrollo del sistema, esto con el fin de evaluar la duración del proyecto.
- Diseño: al llegar a esta etapa significa que se tiene un panorama un poco más claro de lo que el cliente requiere, para ello se harán diseños del sistema a nivel de apariencia, también diseños lógicos como ser diagramas que muestren el flujo que podrá tener la información del sistema de donde se extraerán los datos y adonde se almacenarán.
- Desarrollo: una vez teniendo los diseños lógicos y estéticos que tendrá el sistema se comienza el desarrollo o codificación del sistema.
- Pruebas: esta etapa es donde el usuario y los desarrolladores interactúan al

momento de fijar entregables para poder estar realizando constantes pruebas y de esta manera reducir el tiempo de corrección de errores o una detección temprana de fallas al momento de levantar requerimientos.

- **Implementación:** una vez aprobadas las pruebas por el usuario se comienza la etapa implementación en la cual el sistema se pone a disposición del usuario en un tiempo paralelo con un sistema antiguo o con un proceso manual previo al sistema nuevo para comprobar el funcionamiento de este.
- **Mantenimiento:** esta etapa consiste en la reparación de errores encontrados luego de que el sistema esté en producción o al momento de necesitar nuevas funcionalidades en este sistema.

2.2.2. Micro entorno

No es un misterio que en nuestro país (Honduras) actualmente se está en el auge del uso de tecnologías de información para el control de las actividades de todo tipo en instituciones grandes y negocio ya sean pequeños o medianos, lo que ha dejado como resultado que se deban de tener áreas específicas para la gestión y control de todas estas tecnologías y como es en el caso de las instituciones o empresas más grandes equipos profesionales y de alto nivel técnico los cuales sean capaces no solo de la gestión y el control sino también del diseño e implementación de las mismas.

Centrándonos más en nuestro tema principal, la experiencia que se le brinda al usuario ha tomado más relevancia que nunca y cada día vemos como las instituciones le dedican un mayor esfuerzo a este tema, pero no obstante las áreas de desarrollo no han evolucionado de la mano y estas no van de la mano con las exigencias actuales, lo que ocasiona que durante el “ciclo de vida del desarrollo de software” exista un poco de descontrol dado que como se explicara más adelante en este mismo apartado en nuestro país son muy pocas las instituciones

que cuentan con una sub división dentro del área de desarrollo para equipos de Front-End y Back-End lo que se traduce en colaboradores a los que popularmente se les denomina “todólogos”. Estos colaboradores cumplen las dos funciones y dependiendo el proyecto en que este participe, sus funciones y responsabilidades cambian y se adaptan al desarrollo y en temas de desempeño esto puede ocasionar que los tiempos de entrega se alarguen.

Con lo anteriormente mencionado se realizó una pequeña investigación que abarco 4 instituciones financieras (Bancos, Cooperativas) y 3 instituciones del estado con el propósito de conocer cómo está estructurada el área de desarrollo en diversas instituciones y aún más específicamente como organizar el equipo de Back-End y Front-End, los resultados de esta investigación nos reflejó que de las 7 instituciones 5 no cuentan con una sub división interna entre Back-End y Front-End y sus colaboradores son los encargados de los dos temas y dependiendo el desarrollo en el que se encuentren su jefe directo les asigna las responsabilidades que tendrán en dicho desarrollo no obstante dos de estas si contaban con equipos de trabajo específico para Back-End y Front-End con la característica que ninguna tenía una sub división interna en el área de back-en y aunque en términos generales existe la división internamente no.

Es de esta forma con la que podemos iniciar a tocar el tema de las tecnologías y estándares o políticas que son usados en nuestro país por las empresas (Muestra conformada por 7 instituciones del sector público y privado) que cuentan con areas de tecnología;

1. Estándares: En cuanto a los estándares en nuestro país no pudimos localizar que una institución adoptara estándares exclusivos para el manejo de equipo de Back-End y esto se resumía más a políticas y normas internas y metodología de desarrollo generales (Agiles) pero en las cuales no definían roles específicos para el equipo de Front-End
2. Herramientas: En cuanto a las tecnologías que los equipos o en el caso de las

instituciones que no cuentan con un equipo específico como tal serán sub divididas en dos:

- a. **Editores:** En el tema de los editores poniendo énfasis en que nuestra muestra podría no ser determinante, se pudo observar cómo el 100% de las instituciones utilizan Visual Studio Code.
- b. **Tecnologías:** En este apartado y en caso contrario al anterior si es variado y difiere de institución a institución con algunas excepciones siendo estas tecnologías:
 - i. GenExus
 - ii. .Net
 - iii. Visual Basic
 - iv. Entity
 - v. Angular
 - vi. Wordpress

2.3. Teorías de sustento

Al hablar de sistemas web, los cuales involucran internet, intranet y el desarrollo de los mismos, sería de gran utilidad conocer cuál es el origen de los mismos como han evolucionado y que cambios han provocados en la sociedad y en su diario vivir, asimismo analizar porque se dice que estos cambios han sido tan radicales que ha sido considerado en la tercera y cuarta revolución industrial.

La palabra «revolución» indica un cambio abrupto y radical. Las revoluciones se han producido a lo largo de la historia cuando nuevas tecnologías y formas novedosas de percibir el mundo desencadenan un cambio profundo en los sistemas económicos y las estructuras sociales. (Schwab,2016).

Las cuatro revoluciones industriales que han surgido han cambiado radicalmente la forma en la que las empresas producen y los clientes consumen, ya sean que estemos hablando de productos o servicios. En origen del internet se produce en la tercera revolución industrial, la cual fue conocida como la revolución digital.

La tercera revolución industrial se inició en la década de 1960, Generalmente se la conoce como la revolución digital o del ordenador, porque fue catalizada por el desarrollo de los semiconductores, la computación mediante servidores tipo «mainframe» (en los años sesenta), la informática personal (décadas de 1970 y 1980) e internet (década de 1990). (Schwab,2016)

De la misma forma el Schwab (2016) menciona que la cuarta revolución industrial comenzó a principios de este siglo y se basa en la revolución digital, la cual se caracteriza por un internet más ubicuo y móvil, por sensores más pequeños y potentes que son cada vez más baratos, y por la inteligencia artificial y el aprendizaje de las máquinas.

A diferencia de la tercer y cuarta revolución industrial, la cuarta revolución el ingeniero llamado Klaus Schwab explica que la base si bien es cierto es la tecnología, es más por el lado de la infraestructura de la revolución industrial. (Schwab, 2016)

Y con lo anterior podemos analizar que la tecnología fue una base para realizar la cuarta revolución industrial en la cual, se mejoró y se creó una tecnología, sistemas y productos más eficientes.



Figura 4. Resumen revolución industrial

Fuente: E. (2021)

Por otra parte, cuando hablamos del ciclo de vida del desarrollo de software existen muchos aspectos que podemos tomar en consideración, como ser: El diseño desarrollo las pruebas y el despliegue de una aplicación web. Estos procesos pueden automatizarse utilizando algunas herramientas actuales de integración y despliegue continuo las cuales implementan metodologías DevOps cómo ser Azure server o services.

La gestión del software es una práctica más amplia, que abarca las actividades continuas de diseño, prueba, funcionamiento y mejora de las aplicaciones de software para que sigan facilitando la creación de valor. Los componentes de software pueden evaluarse continuamente utilizando un ciclo de vida que sigue el componente desde su ideación hasta su desarrollo. (AXELOS, 2019)

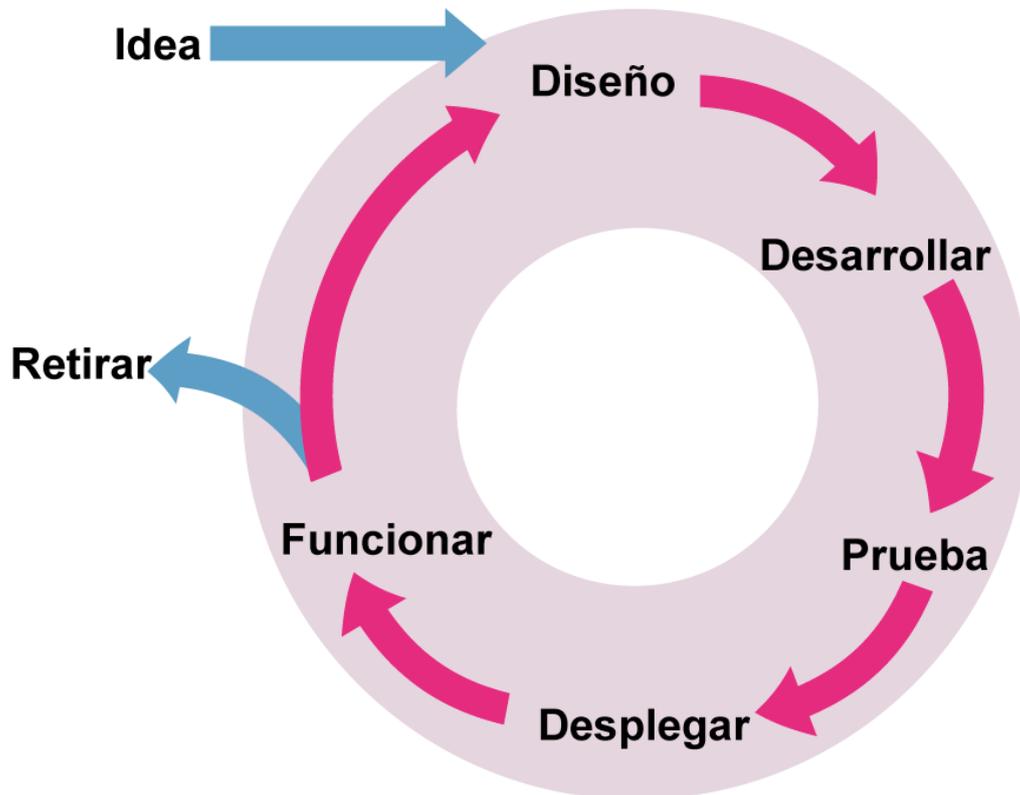


Figura 5. Ciclo de vida del desarrollo de software según ITIL.
Fuente: (AXELOS, 2019)

Los equipos de desarrollo de software ágil se centran a menudo en la entrega rápida de incrementos de producto a expensas de una visión más holística que tenga en cuenta la operatividad, la fiabilidad y la capacidad de mantenimiento de estos productos. (AXELOS, 2019).

En este punto es donde el papel del componente de Front-End juega un papel vital y de muy gran importancia, el cual es poder incorporarlo desde las etapas tempranas del ciclo de vida del desarrollo de software, para poder interpretar de manera gráfica la visión que se tiene para un aplicativo web.

2.3.1. Análisis de las metodologías

Luego del análisis de ciertas metodologías al igual que estándares internacionales se llegó a la decisión de utilizar la metodología ITIL en sus versiones 3 y 4, metodología la cual ha sido estudiada a lo largo de la maestría de Gestión de Tecnologías de la Información en UNITEC, además del estándar internacional ISO/IEC 12207 el cual es un marco enfocado en los procesos del ciclo de vida del desarrollo de software.

ITIL es un conjunto de buenas prácticas enfocadas en la gestión de los servicios ofrecidos por el área de TI, que a su vez nos brindan una serie de procesos los cuales tienen el objetivo de ayudar a la organización a mejorar la calidad de los servicios que ofrece dentro como fuera de la organización, a los cuales ayuda constantemente a estos ligados a las metas.

ITIL® es un marco de trabajo público que describe las mejores prácticas en la gestión de Gestión de Servicios de TI (ITSM). De hecho, es el marco de trabajo más utilizado en el mundo para la ITSM, proporcionando un marco para el gobierno de TI, y la gestión y control de los servicios de TI. (Itil Qué es y Breve Historia, 2020)

www.ibm.com (2019) nos menciona que una de las principales metas de ITIL es asegurar que los servicios de IT estén alineados con los objetivos del negocio, aun cuando los objetivos del negocio cambien.

Dándonos a conocer que es un objetivo primordial para ITIL el reforzar y ayudar a mejorar los servicios que se ofrecen pero que al igual estos puedan adaptarse y acoplarse aun cuando estos últimos cambien situación que es muy común en nuestro ambiente.

La calidad del servicio está determinado por una serie de factores las cuales pueden tener similitudes entre organizaciones o ser completamente distinto, esto también es

influenciado por el ambiente como por los clientes (personas internas o externas a las cuales hacen uso del servicio), siendo uno de estos factores la velocidad o eficiencia con la cual se es finalizado o entregado el servicio, tomando en cuenta esto último se responderá por qué se ha decidido hacer uso de esta metodología dado que esta tiene definido de forma muy clara un ciclo de vida el cual está compuesto por 5 fases:

- La estrategia
- El diseño
- La transición
- La operación
- Mejora continua

Etapas que en su conjunto tienen la finalidad la gestión del servicio para que estos funcionen de forma eficiente, los cuales pueden evolucionar continuamente para poder seguir progresando y dando los mejores resultados en todo momento; dos puntos importantes a tomar en consideración es que, aunque si nos presentan 5 fases éstas son un todo y no deben de ser vistas como tareas independientes una de la otra a lo que se le conoce como visión holística y segundo no debemos de olvidar que el alineamiento de nuestros servicios con los objetivos del negocio es de vital importancia y debemos de mantener estos en todo momento.

La normativa ISO/IEC/IEEE 12207 es un marco de trabajo el cual se centra en los procesos que componen o conforman el ciclo de vida de un sistema, pudiendo ser desde su adquisición o desarrollo hasta cuando este es dado de baja o remplazado.

<https://www.iso.org/> nos menciona

ISO/IEC/IEEE 12207:2017 also provides processes that can be employed for defining, controlling, and improving software life cycle processes within an organization or a project.

Dejándonos aún más claro el propósito principal de esta normativa.

Al igual que ITIL con sus 5 fases del ciclo de vida la normativa ISO ISO/IEC/IEEE 12207 cuenta con cuatro fases las cuales componen el ciclo de vida del software, estas fases son las siguientes:

- Agreement processes
- Organizational project-enabling processes
- Technical management processes
- Technical processes

Los cuales a su vez tienen una serie de sub divisiones las cuales se encargan de detallar a fondo de tal forma que podemos tener un panorama claro de cómo se compone cada una de estas fases, la importancia de estas y su función.

2.3.2. Antecedentes de las metodologías

Los inicios de ITIL se pueden remontar a los años 1980, y como es muy común en nuestro ambiente nace producto de una necesidad o deficiencia la cual detecto el gobierno británico, debido a que la rama del gobierno encargada de las TI se estaba volviendo cada vez más indispensable e importante para el buen funcionar del mismo, por lo cual decidieron que en un futuro cercano sería fundamental el poder tener un vasto control y gestión de estos servicios que la rama de TI les ofrecía y es de esta forma donde nace ITIL V1, la cual para esa primera versión constaba de 10 libros que posteriormente se convirtieron en 40, y de esta forma hasta la actualizada se ha continuado con la tendencia de actualizar y revisar cada una de las versiones siendo estas en el 2000 ITIL V2, en el 2007 ITIL V3, en el 2011 se dio a conocer ITIL V3 2011 la cual será de mucha importancia en la presente investigación debido a que es aquí donde se decanta por el ciclo de vida y nos presenta por primera vez las 5 famosas fases que proponen y así podemos pasar a ITIL V4 en el año 2019.

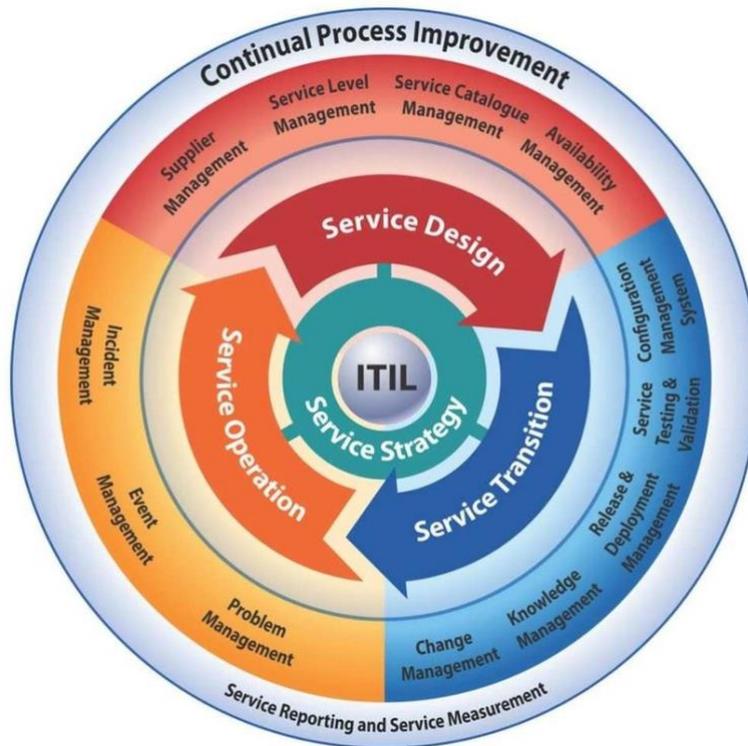


Figura 5. Ciclo de vida del servicio según ITIL.

Fuente: (ITIL V3.2)

Los inicios de la normativa ISO/IEC/IEEE 12207 se remontan a los años 1995 cuando esta fue presentada por primera vez, siendo uno de los primeros estándares el cual nos presentaba un conjunto de procesos detallados al igual que explicados los cuales eran requeridos para el mantenimiento y el desarrollo de sistemas de información, pudiendo tener un mayor de talla de cada una de las salidas y recursos necesarios para cada uno de los procesos que se detallan.

Desde 1995 hasta la fecha han sido publicadas 5 versiones del estándar incluyendo la versión 2017 la cual ha sido la última en ser publicada.

2.3.3. Análisis crítico de las metodologías

La selección de ITIL V3 y ITIL V4 para la presente investigación debido a la amplia gama de prácticas las cuales son de vital importancia siendo una de estas gestión y desarrollo de software al igual que una serie de ciclos los cuales son indispensables y en los que se basara

nuestra investigación como ser el ciclo de vida del servicio y el ciclo de vida de desarrollo.

La razón por la cual se hará uso de dos versiones de la misma metodología es debido al enfoque de estas y aunque se podría interpretar que una versión viene a sustituir la otra esto no es del todo cierto debido a que aunque mejoran, actualizan o renuevan metodologías o temas dependiendo la actualizad en ocasiones dan por asumidos criterios que se han venido manejando y no son incluidos por no hacer aún más grande o engorrosos los libros siendo, tomando de ejemplo el enfoque más explicativo y detallado de ITIL V3 en cuanto a la gestión y sus componentes y ya un ITIL V4 más orientado a la generación de valor el cual no elimina pero si da por asumido los conceptos anteriores siendo un claro ejemplo el nivel de detalle que se obtienen en ITIL V3 de las 5 fases, como se mostrara a continuación:

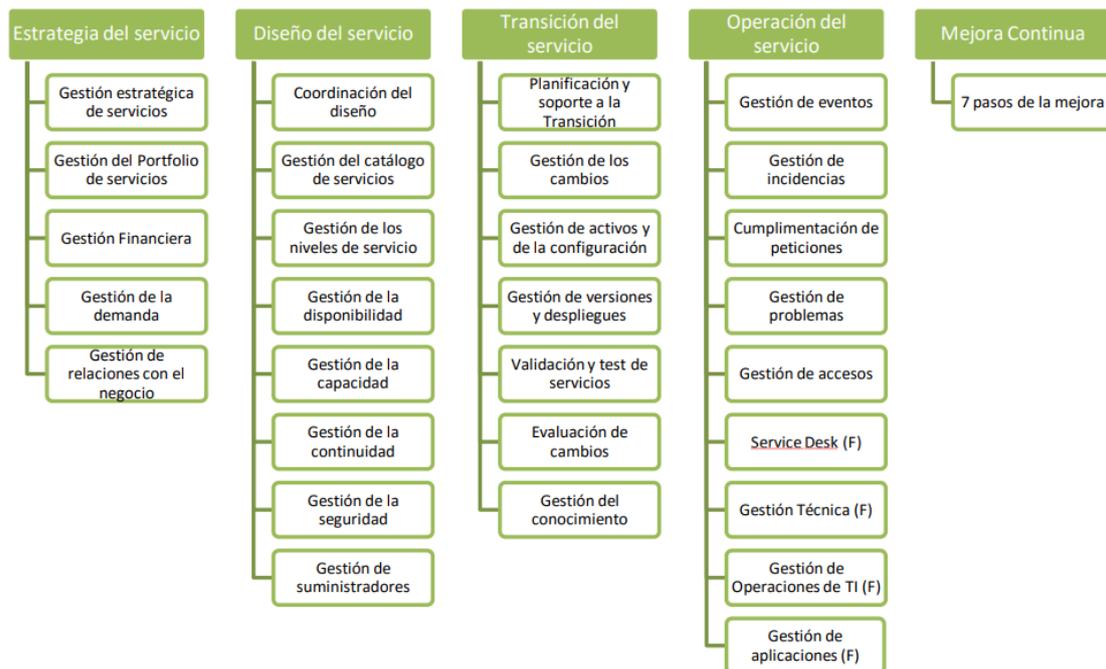


Figura 6. Cinco fases de ITIL.

Fuente: (ITIL V3)

Se decidió la introducción de la normativa ISO/IEC/IEEE 12207 primeramente por el gran reconocimiento que la organización internacional ISO tiene y como segundo punto debido a que esta normativa nos presenta su propio ciclo de vida del desarrollo de software sin entrar en

detalles en el cómo, lo que nos abre las puertas a un sinfín de información y buenas practicas las cuales podemos ajustar y tomar la información que más nos será de importancia.

2.4. Herramientas

El uso de herramientas tecnológicas abre un sinfín de posibilidades que van desde agilidad, eficiencia, mejores resultados, calidad, comodidad y mayor satisfacción de nuestros clientes, para fines de conseguir los resultados esperados y llegar a la meta en la presente investigación se hará un clasificación la cual está compuesta por dos divisiones principales de Herramientas, herramientas técnicas y herramientas para el estándar, dado que para lograr mejores tiempos en el ciclo de vida de desarrollo es necesario dar a los colaboradores las mejores herramientas técnicas para que ellos puedan desarrollar su trabajo de forma eficiente y también hacer uso de herramientas las cuales permitan gestionar bien el ciclo de vida en cada una de sus etapas y como mencionamos anteriormente en la presente investigación se hará uso de ITIL.

A continuación, se detallan las herramientas seleccionadas con su respectiva clasificación

1. Técnicas

- a. Figma: Herramienta que será utilizada por nuestros colaboradores a la hora de realizar todos sus diseños, la cual les permitirá de una forma más rápida y brindando mejores resultados esta etapa inicial del proceso de desarrollo.
- b. Azure DevOps: esta será la herramienta mediante la cual se logrará que todos los colaboradores destinados o encargados de una tarea de desarrollo puedan trabajar en forma conjunta de una forma muy intuitiva y cómoda, permitiéndoles mantener un control de modificaciones y versiones controlado.

2. Metodológicas

- a. Trello: será la encargada o la usada para el control, la gestión y organización del proyecto, esta será la encargada para el desarrollo de la idea principal como

los puntos a seguir, hasta definir las actividades finales que conformaran el desarrollo, estas se distribuyen entre el total de los colaboradores.

- b. Slack: esta será la herramienta encargada para mantener el canal de comunicación usado para mantener una gestión oportuna y a tiempo de las actividades, así como fungir como un gestor de conocimiento en el cual se podrán integrar el total de documento que sean requeridos para realizar toda actividad planeada.

2.5. Marco legal

2.5.1. Reglamento sobre el gobierno electrónico

Al hablar del desarrollo de aplicaciones web internas o externas de una institución gubernamental, debe someterse a los principios y procedimiento que rigen las relaciones de los particulares con las entidades de la administración pública. El Reglamento sobre el gobierno electrónico, tiene como finalidad definir estos principios, desde las acciones que deberían tener los desarrollados, hasta como debe ser el proceso y priorización para la contratación de servicios en la nube, consecuencia de esto identificamos los relevantes para el desarrollo de aplicaciones web los artículos siguientes:

Artículo 28.- El Despacho de Gestión e Innovación Pública mantendrá directorios actualizados de las aplicaciones que desarrollen los órganos de la Administración Pública para su libre reutilización, especialmente en aquellos campos de especial interés para el desarrollo de la administración electrónica y de conformidad con lo que al respecto se establezca en el esquema de interoperabilidad conforme a lo establecido en el presente Reglamento y su normativa derivada. (Reglamento Sobre Gobierno Electrónico, 2020)

Artículo 29.- Los órganos de la Administración Pública privilegiará el desarrollo, contratación o adquisición de licenciamiento de software libre, para la creación de sistemas. Para tal efecto, se podrán utilizar las aplicaciones y herramientas que cumplan con los siguientes criterios: 1. Garantizar las mejores condiciones en cuanto a precio, calidad, financiamiento, oportunidad y demás circunstancias pertinentes; y, 2. Ser actualizable y escalable en sus funciones. (Reglamento Sobre Gobierno Electrónico, 2020)

Artículo 30.- Los Órganos de la Administración Pública deberán de privilegiar el uso y desarrollo de software libre, sobre software privativo, en programas, aplicaciones y/o sistemas informáticos, y se deberán apegar a los siguientes principios de software libre:

1. Libertad para ejecutar el programa independientemente de su propósito;
2. Acceso al código fuente, que permita estudiar el funcionamiento del programa y adaptarlo a las necesidades específicas;
3. Libertad para redistribuir copias entre los mismos Órganos de la Administración Pública, y
4. Libertad para mejorar el programa y publicarlo para toda la Administración Pública.

(Reglamento Sobre Gobierno Electrónico, 2020)

Artículo 31. - Los sistemas informáticos de tramitación de la Administración Pública deberán ser actualizados permanentemente con el objeto de permitir su correcto funcionamiento y la más fluida y expedita interconexión e interoperabilidad entre sí y con otras instituciones públicas. (Reglamento Sobre Gobierno Electrónico, 2020)

Artículo 32. - Cada uno de los Órganos de la Administración Pública, en el ámbito de sus respectivas competencias, propiciará la capacitación de los servidores públicos a su cargo en materia de tecnologías de la información y comunicaciones. (Reglamento Sobre Gobierno

Electrónico, 2020)

Artículo 54.- Responsabilidad en el manejo del usuario interno o manipulación del sistema: Será responsabilidad exclusiva de cada Servidor Público que esté autorizado al manejo o manipulación del sistema y de los datos, el manejo de su contraseña privada de acceso a la plataforma digital. Queda prohibido compartir la misma con terceros. La instrucción recibida por parte de un superior jerárquico para usurpar dicha facultad de acceso no es eximente de responsabilidad. Los sistemas deberán registrar los accesos de forma segura. La contravención a esta disposición dará lugar a iniciar el procedimiento disciplinario que corresponda, y se procederá a poner en conocimiento a la autoridad competente para que esta determine si se ha incurrido en responsabilidad administrativa, civil o penal. (Reglamento Sobre Gobierno Electrónico, 2020)

Artículo 56.- Las instituciones serán las responsables por la seguridad de la información que manejan en sus bases de datos y deben observar, implementar y operar bajo los criterios generales de seguridad de la información contenidos en la Ley de Transparencia y Acceso a la Información Pública.

Artículo 59.- Las instituciones, previo al inicio de la puesta en operación de un aplicativo de cómputo, realizarán el análisis de vulnerabilidades correspondiente, el cual será realizado por un tercero, distinto a quien desarrolló el aplicativo. El resultado del análisis deberá custodiarse de manera reservada para efectos de eventuales comprobaciones sobre la seguridad de la información o auditorías. (Reglamento Sobre Gobierno Electrónico, 2020)

Artículo 62.- El funcionario o empleado público que por cualquier medio sustraiga, borre o permita que un tercero sustraiga o borre información contenida en una base de datos o en los sistemas informáticos o equipo de cómputo del Estado, incluyendo el que le hubiere sido asignado para el desempeño de sus funciones, es civil y administrativamente responsable por

los daños y perjuicios causados en contra de la Administración Pública o de terceros afectados, sin perjuicio de la responsabilidad penal que pudiere originarse de sus actos. (Reglamento Sobre Gobierno Electrónico, 2020)

2.5.2. Código penal

El Código Penal de Honduras contempla en el Título XXII Seguridad de las redes y de los sistemas informáticos, el cual no menciona específicamente como realizar el desarrollo de un sistema web, pero sí en que delitos podemos incurrir, por ejemplo: al crear sistemas informáticos con el objetivo de acceder de manera ilícita a la información, al hacer uso de sistemas sin un acceso autorizado, entre otros delitos. Estos artículos son:

ARTÍCULO 398.- ACCESO NO AUTORIZADO A SISTEMAS INFORMÁTICOS.

Debe ser castigado con pena de prisión de seis (6) a dieciocho (18) meses o multa de cien (100) a doscientos (200) días quien, vulnerando las medidas de seguridad establecidas para impedirlo, accede sin autorización a todo o en parte de un sistema informático. La pena del párrafo anterior se debe aumentar en un tercio (1/3) si el sistema al que se accede se refiere a estructuras o servicios esenciales para la comunidad. (Código Penal de Honduras, 2021)

ARTÍCULO 399.- DAÑOS A DATOS Y SISTEMAS INFORMÁTICOS.

Quien por cualquier medio y sin autorización introduce, borra, deteriora, altera, suprime o hace inaccesible de forma grave datos informáticos, debe ser castigado con la pena de prisión de uno (1) a dos (2) años o multa de cien (100) a trescientos (300) días. Quien sin estar autorizado inutiliza, total o parcialmente, el funcionamiento de un sistema informático, impidiendo el acceso al mismo o imposibilitando el desarrollo de alguno de sus servicios, debe ser castigado con la pena de prisión de uno (1) a tres (3) años o multa de cien (100) a cuatrocientos (400) días. Se debe aumentar en un tercio (1/3) las penas señaladas en los dos (2) párrafos anteriores, cuando las conductas descritas causen un grave daño económico o afecten a estructuras o servicios esenciales para la comunidad. (Código Penal de Honduras, 2021)

ARTÍCULO 400.- ABUSO DE DISPOSITIVOS. La fabricación, importación, venta, facilitación o la obtención para su utilización de dispositivos, programas informáticos, contraseñas o códigos de acceso, destinados o adaptados para la comisión de los delitos de daños informáticos o de acceso ilícito a sistemas informáticos, debe ser castigada con la pena de prisión de seis (6) meses a un (1) año o multa de cien (100) a doscientos (200) días.

ARTÍCULO 401.- SUPLANTACIÓN DE IDENTIDAD. Debe ser castigado con la pena de prisión de seis (6) meses a un (1) año o multa de cien (100) a trescientos (300) días, quien con ánimo defraudatorio y a través de las tecnologías de la información y la comunicación, suplanta la identidad de una persona natural o jurídica. (Código Penal de Honduras, 2021)

ARTÍCULO 402.- CIRCUNSTANCIAS AGRAVANTES. Se deben aumentar en un tercio (1/3) las penas previstas en este título, quienes ejecuten el delito concurriendo alguna de las circunstancias siguientes:

- 1) El culpable es el responsable o encargado del sistema o, cuenta con autorización para acceder a los datos o sistemas informáticos; o
- 2) El culpable pertenece a un grupo delictivo organizado.

En los delitos previstos en este título, junto a la pena correspondiente, se debe imponer la inhabilitación especial por el mismo período para el ejercicio de profesión, oficio, industria o comercio relacionados con la conducta. Si el delito ha sido cometido por funcionario o empleado público también se debe imponer, en las mismas condiciones que en el caso anterior, la pena de inhabilitación absoluta durante el tiempo de la condena. (Código Penal de Honduras, 2021)

ARTÍCULO 404.- REGLAS ESPECIALES DE JURISDICCIÓN. Los Órganos Jurisdiccionales nacionales deben conocer de los delitos informáticos, cuando se ejecuten en los casos siguientes: 1) En Honduras, aunque se dirijan contra datos o sistemas informáticos

situados fuera de éste; o, 2) Contra datos o sistemas informáticos situados en Honduras, aunque el culpable hubiese actuado desde fuera del territorio nacional. (Código Penal de Honduras, 2021)

ARTÍCULO 405.- DEFINICIONES LEGALES. A los efectos de este título, se entiende por:

1) Datos Informáticos: las unidades básicas de información, cualquiera que sea su contenido, expresados en una forma que permita su tratamiento por un sistema de información, incluyendo los programas que hacen posible que esta función se lleve a cabo;

2) Sistema Informático: un dispositivo o conjunto de dispositivos interconectados o relacionados entre sí, que permiten, gracias a un programa, el tratamiento automatizado de datos informáticos, de manera que abarca tanto el hardware como el software necesario para su funcionamiento; y,

3) Programa Informático: la secuencia de instrucciones o indicaciones necesarias para que el sistema informático pueda realizar una función o una tarea u obtener un determinado resultado. (Código Penal de Honduras, 2021)

CAPÍTULO III. METODOLOGÍAS DE LA INVESTIGACIÓN

3.1 Tipo de enfoque

El tipo de enfoque que se utiliza en esta investigación es cualitativo puesto que se basan en explorar, descubrir, indagar, recolectar puntos de vista, experiencia, las cuales permiten identificar características, hipótesis, replantear procesos y obtener resultados que permitan afianzar o crear una perspectiva teórica.

De esta manera el investigador cualitativo utiliza técnicas para recolectar datos, como la observación no estructurada, entrevistas abiertas, revisión de documentos, discusión en grupo, evaluación de experiencias personales, registro de historias de vida, e interacción e introspección con grupos o comunidades. (Hernández Sampieri et al., 2014).

En la búsqueda cualitativa, en lugar de iniciar con una teoría y luego “voltear” al mundo empírico para confirmar si ésta es apoyada por los datos y resultados, el investigador comienza examinando los hechos en sí y en el proceso desarrolla una teoría coherente para representar lo que observa (Esterberg, 2002).

3.2 Alcance

El objetivo principal de esta investigación añade un alcance descriptivo, puesto que se tendrá que describir las diferentes herramientas, procesos, roles, uso de los roles, para analizar el impacto que tiene el Front-End en el ciclo de vida del desarrollo de software. Se tendrá que identificar los momentos en los cuales el Front-End debe intervenir, apoyar o limitarse en sus actividades en el ciclo de vida y de esta manera de comprobar el impacto de involucrar al Front-End.

Este tipo de investigaciones se propone describir el comportamiento de variables y/o

identificar tipos o pautas características resultantes de las combinaciones de un cierto número de ellas. Las investigaciones descriptivas se ocupan entonces de identificar las variables relevantes del objeto o asunto investigado y luego de averiguar cómo se comportan dichas variables. (Ynoub, 2008)

3.3 Población

La población de este estudio estuvo integrada por 12 personas del equipo de desarrollo de software interno, de la sección de proyectos de banca central, del departamento de tecnología del Banco Central de Honduras entre los años 2018 al 2022, esto debido a que esta población participó en la transición de una arquitectura de desarrollo tradicional a la nueva arquitectura en la cual se definen los componentes y equipos de desarrollo para Back-End y Front-End, asimismo, toda solución de tecnología e información de tipo software en la arquitectura de micro servicios en Back-End y que utilicen el Framework de Angular en Front-End.

3.4 Muestra

En esta investigación se utiliza una técnica de muestreo **no probabilístico**, específicamente una muestra de expertos debido a que la elección de elementos para que formen parte de la muestra no se basa en una elección por probabilidad o aleatoria, sino que se deben seleccionar una muestra dirigida y seleccionando un grupo de expertos, según sus características y que estas estén relacionadas con los objetivos de esta investigación.

En las muestras no probabilísticas, la elección de los elementos no depende de la probabilidad, sino de causas relacionadas con las características de la investigación o los propósitos del investigador (Johnson, 2014, Hernández-Sampieri et al., 2013 y Battaglia,

2008b)

Asimismo, en ciertos estudios es necesaria la opinión de individuos expertos en un tema. Estas muestras son frecuentes en estudios cualitativos y exploratorios para generar hipótesis más precisas o la materia prima del diseño de cuestionarios. (Hernández Sampieri et al, 2014)

3.5 Unidad de análisis

La unidad de análisis para esta investigación está representada por Software y las etapas para el desarrollo del mismo.

3.5.1 Criterios

2.2.2.7 Criterios de inclusión

- Proyectos asignados a la sección de proyectos de banca central
- Proyectos que implementen metodologías ágiles
- Los que implementan componentes de Back-End y Front-End separado
- Las que son aplicaciones Web

2.2.2.8 Criterios de exclusión

- Las aplicaciones desarrolladas con metodologías tradicionalmente
- Proyectos ya iniciados por otras secciones
- Aplicaciones con arquitecturas monolíticas

3.6 Operacionalización de variables

Tabla 4. Operacionalización de variables

Variables	Dimensiones	Indicadores	Ítems
Personal humano	Programador Full-Stack	Programadores que realizan todas las actividades de programación y análisis de sistemas.	Programadores que pasan sobre cargados de trabajo ya que realizan actividades de todo el ciclo de vida del desarrollo de software.
	Programador especializado	Programadores especializados en áreas como: Front-End, Back-End y análisis de sistemas	Programadores que se especializan en un área lo que aumenta la eficiencia y reducción en los tiempos en que realizan sus actividades.
Clima motivacional	Satisfacción del cliente	Reducción en solicitudes de cambio	Los usuarios solicitan cambios muy frecuentemente, incluso cuando un sistema es reciente debido a que no satisface sus necesidades

	Motivación del equipo de desarrollo	Aumento de entregas en tiempo y forma	El equipo se siente ofuscado por la cantidad de trabajo, lo cual influye en retrasos en la entrega de sistemas.
Carga laboral	Área de negocio	Reuniones recurrentes	Participar en reuniones con una mayor frecuencia para analizar requerimientos, evaluar componentes y confirmar el buen funcionamiento de los sistemas, obliga al usuario a involucrarse más en el ciclo de vida del desarrollo de software.
	Área de tecnología	Cantidad de tareas	Desarrollar prototipos, reuniones para modificaciones, constantes ajustes a los diseños y prototipos son algunas de las tareas adicionales que tendrá que realizar el equipo.

3.7 Instrumentos

Al tomar en consideración que esta investigación se basa en las experiencias y conceptos de expertos en el área de desarrollo y afines a esta, debido a que cada organización tiene su forma de adaptar dependiendo sus características y recursos su funcionamiento, por lo cual será de vital importancia poder conocer las experiencias de este personal al igual que sus puntos de vista por tal motivo se ha definido como instrumento para recolección de datos las entrevistas semiestructuradas y la consulta a expertos ya que estas tienen la libertad de introducir preguntas para definir mejor obtener mayor información de algunos conceptos o situaciones.

3.7.1 Entrevistas semiestructuradas

Según Hernández Sampieri et al (2014) las entrevistas semiestructuradas se basan en una guía de asuntos o preguntas y el entrevistador tiene la libertad de introducir preguntas adicionales para precisar conceptos u obtener mayor información. Es debido a estas características anteriormente mencionadas que se ha decidido la incorporación de este instrumento, debido a que en estas sesiones mediante una guía base la cual contendrá preguntas de puntos específicos las cuales estarán diseñadas para tocar cada punto relevante y de esta forma extraer la mayor cantidad de información, pero no obstante servirán como una guía, no se estará limitando la misma a contestar cada una de estas en el orden sino mas bien serán meramente orientativas y a medida que el entrevistado de su respuesta se podrán ir tomando puntos importantes con los cuales se redactaran en el momento nuevas preguntas para ahondar mas en temas que el entrevistado toque y es de esta forma que se podrá extraer la mayor cantidad de información.

3.7.2 Consulta a expertos

Reguant Alvarez, Mercedes Torrado Fonseca (2016) nos mencionan que es una técnica de recogida de información que permite obtener la opinión de un grupo de expertos a través de la consulta reiterada. Esta técnica, de carácter cualitativo, es recomendable cuando no se dispone de información suficiente para la toma de decisiones o es necesario, para nuestra investigación, recoger opiniones consensuadas y representativas de un colectivo mayor.

El objetivo de esta técnica es el poder maximizar cualquier oportunidad en la cual se disponga de la opción de poder dialogar sobre puntos específicos de la investigación con personal que cuenta con vasta experiencia en el área, en un formato libre donde no se cuenta con una guía o una base lo cual permitirá que la plática fluya en la cual podremos extraer puntos de vistas, consejos, experiencias y todo tipo de información la cual ya sea los estudios o la práctica le han brindado a la persona seleccionada.

3.7.3 Fuentes de información

Las fuentes de información pueden definirse como todo aquel instrumento mediante el cual se tienen acceso a la información que es y será relevante para alcanzar los objetivos planteados durante la investigación. Durante nuestra investigación fueron estos instrumentos los que nos permitieron darnos cuenta del estado actual en el que se encuentra el área que deseamos estudiar.

3.8 Plan de análisis

El análisis de datos en esta investigación se realizó siendo observadores de como los equipos de desarrollo evolucionan producto a las exigencias del desarrollo y clientes, lo que permitió recolectar datos de gestores de proyecto, desarrolladores y usuarios finales, posteriormente analizar y extraer e identificar patrones, con el fin de generar resultado que puedan afianzar la teoría de que tiene un impacto positivo involucrar a Front-End en el ciclo de vida del desarrollo de software. Aunque para esto debamos regresar a etapas previas (recolección de los datos, tareas analíticas y resultados) como lo menciona Hernández Sampieri et al (2014) que el proceso de análisis de datos en una investigación descriptiva incluye tres acciones progresivas (vertientes) en paralelo íntimamente ligadas: recolección de los datos, tareas analíticas y resultados. Las flechas en dos sentidos implican que podemos regresar a etapas previas y las flechas sin direccionalidad indican actividades asociadas.

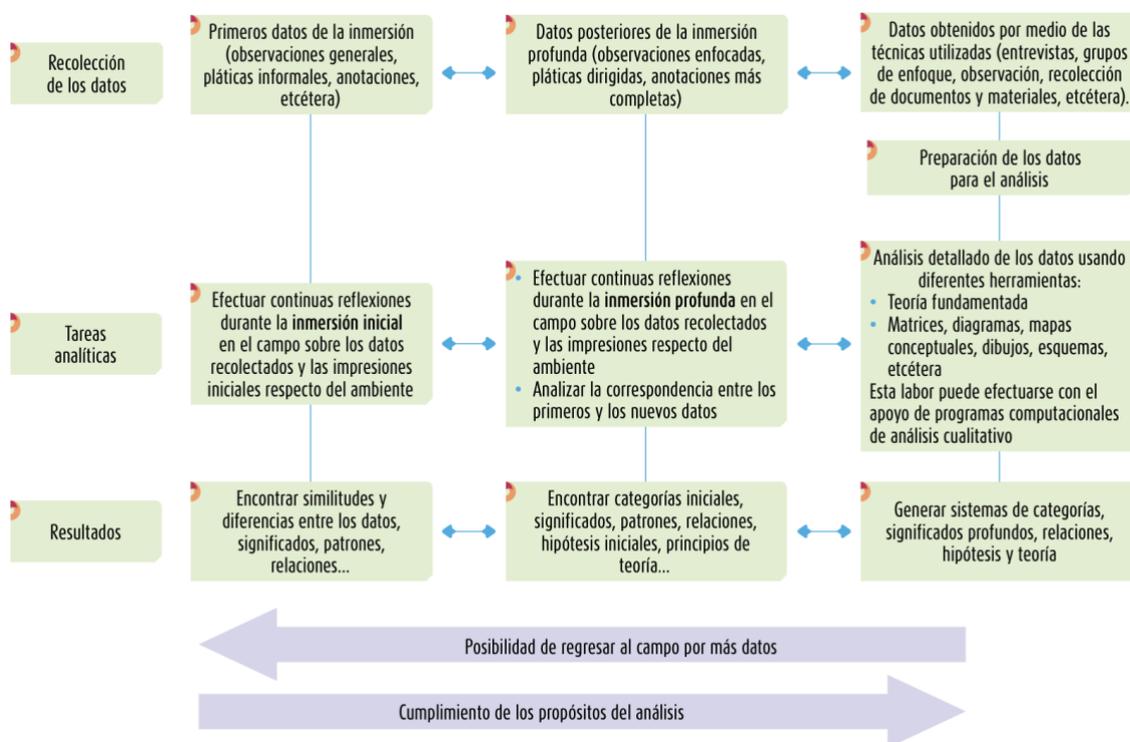


Figura 7. Propuesta de “coreografía” del análisis cualitativo

Fuente: Hernández Sampieri et al, 2014

3.9 Matriz metodológica

Tabla 5. Matriz metodológica

Fuente: Propia

Pregunta general	Objetivo General	Preguntas específicas	Objetivos específicos	Metodología	Variables	Indicadores	Instrumento
¿Involucrar al equipo de Front-End desde el inicio del ciclo de vida de desarrollo del software del BCH, mejorará a eficiencia en los tiempos de entrega del	Evaluar el ciclo de vida del desarrollo de software del área de desarrollo del BCH,	¿El ciclo de vida de desarrollo de un sistema utilizado por el equipo de Front-End es el adecuado para poder acoplar o integrar los cambios que se proponen?	Identificar que componentes pueden estar afectando la eficiencia, ciclo de vida del desarrollo de software del área de desarrollo del BCH.	Cualitativa	Carga laboral	Cantidad de tareas asignadas	Consulta a experto
	para mejorar la eficiencia en los	¿La implementación de roles en el equipo de desarrollo de Front-End del BCH, tendrá el impacto suficiente en cuanto a tiempos de entrega a la hora de afrontar un desarrollo?	Describir como el uso de roles en el equipo de Front-End puede reducir los tiempos de	Cualitativa	Personal humano	Aumento de entregas en tiempo y forma	Entrevista

software y la satisfacción del cliente?	tiempos, utilizando roles en el personal de desarrollo de Front-End.	desarrollo en los proyectos.				
		Definir roles y sus funciones para el equipo de desarrollo del Front-End del BCH.				
	¿Dada la situación actual del equipo de desarrollo de Front-End podría de cierta forma ser contraproducente el alterar como ellos están acostumbrados a realizar su trabajo, o como estos cambios se integran con los demás equipos que tienen relación con este?	Analizar e identificar las etapas del ciclo de vida del desarrollo de software en las cuales la participación del equipo de	Cualitativa	Clima motivacional	Cantidad de opiniones a favor	Entrevista

			desarrollo del Front-End produzca una mejoría.				
--	--	--	---	--	--	--	--

CAPÍTULO VI. RESULTADOS Y ANÁLISIS

En este capítulo se muestran de manera gráfica los datos obtenidos mediante diagramas, tablas y evidencias. Estos datos fueron recolectados de la unidad de muestreo mediante entrevistas semi-estructuradas y consulta a expertos. Con estos datos se formuló un análisis que explique y demuestre el comportamiento de los patrones observados. Asimismo, analizaremos y empezaremos detallando el ciclo de vida del desarrollo de un sistema.

En la primera sección de este capítulo se detalla el proceso actual en el ciclo de vida del desarrollo de un software, con la intervención desde el equipo de Front-End. Posteriormente se muestra el resultado y análisis a profundidad de los datos recolectados con los distintos instrumentos.

4.1 Proceso actual

Esta sección se refiere al proceso del ciclo de vida del desarrollo de software que utiliza la sección de proyectos de banca central en el BCH.

Actualmente la creación de un proyecto de software consiste en múltiples etapas, en las cuales las primeras etapas sólo involucran al personal administrativo, analizando temas como presupuestos, tiempos, necesidades, personal, beneficios, entre otros, pero los desarrolladores comienzan a saber de un nuevo sistema hasta que entran a la etapa de análisis del sistema.

Hasta este punto el personal de Front-End no ha tenido ninguna intervención en este ciclo, el programador de Front-End interviene hasta la etapa de desarrollo que es en la única etapa en la que tiene tareas que realizar, en esta etapa al desarrollador de Front-End se le entrega un diseño creado con herramientas como Pencil, PowerPoint, Paint o Word.

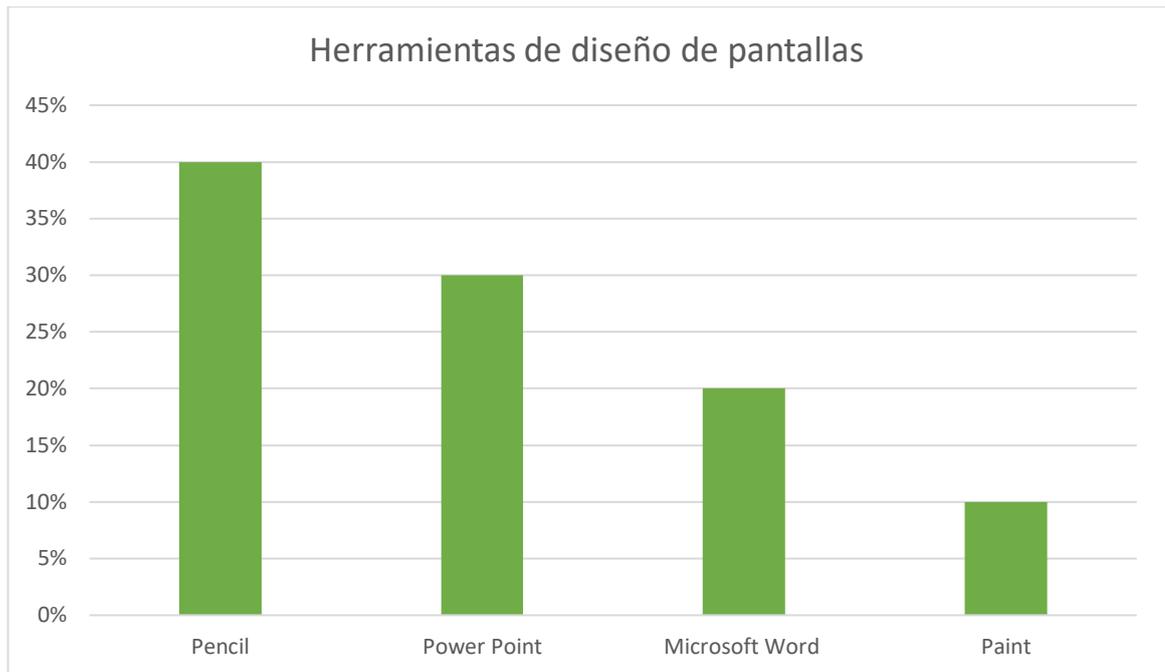


Figura 8. Herramientas de diseño de pantallas.

Fuente: Propia

Éstos diseños contienen la apariencia que los analistas y los usuarios finales o clientes esperan obtener en el sistema, pero estos diseños no son nada detallados sólo contienen información que indica dónde debe ir a una tabla o donde posicionar un botón, pero carecen de información como tamaños de las tablas de los botones, colores, forma, el flujo que tendrá la aplicación y esto provoca frecuentes solicitudes de gestión de cambio, lo que a su vez incurre en retrasos en los cronogramas, entregas retrasadas de productos, poca satisfacción del cliente, aumento de presupuesto, entre otros factores negativos en el proyecto.

Las herramientas de diseño como ser: Pencil, PowerPoint, Microsoft Word, Paint son herramientas utilizadas para diseñar pantallas, aunque el propósito inicial de estas herramientas no es ese, ejemplo Microsoft Word es una aplicación que ayuda a la creación, modificación de documentos, aunque esta herramienta pueda servir para crear imágenes no es su propósito o función principal, a diferencia de Figma, la cual es una aplicación diseñada y creada para diseñadores de aplicaciones web en la cual se encuentran distintas funcionalidades que ayudan

a crear diseños más exactos y detallados con el fin de ofrecer diseños reales de como la apariencia de un sistema podría ser la que recibirá el usuario al finalizar el proyecto.

Aunque también en esta sección están utilizando herramientas o aplicaciones las cuales fueron Creadas para el diseño de aplicaciones como ser Pencil, pero estas aplicaciones son un poco antiguas, por no decir obsoletas y en la **figura 9** podemos observar como sus diseños son poco realistas a lo que tendremos como producto final en un sistema.

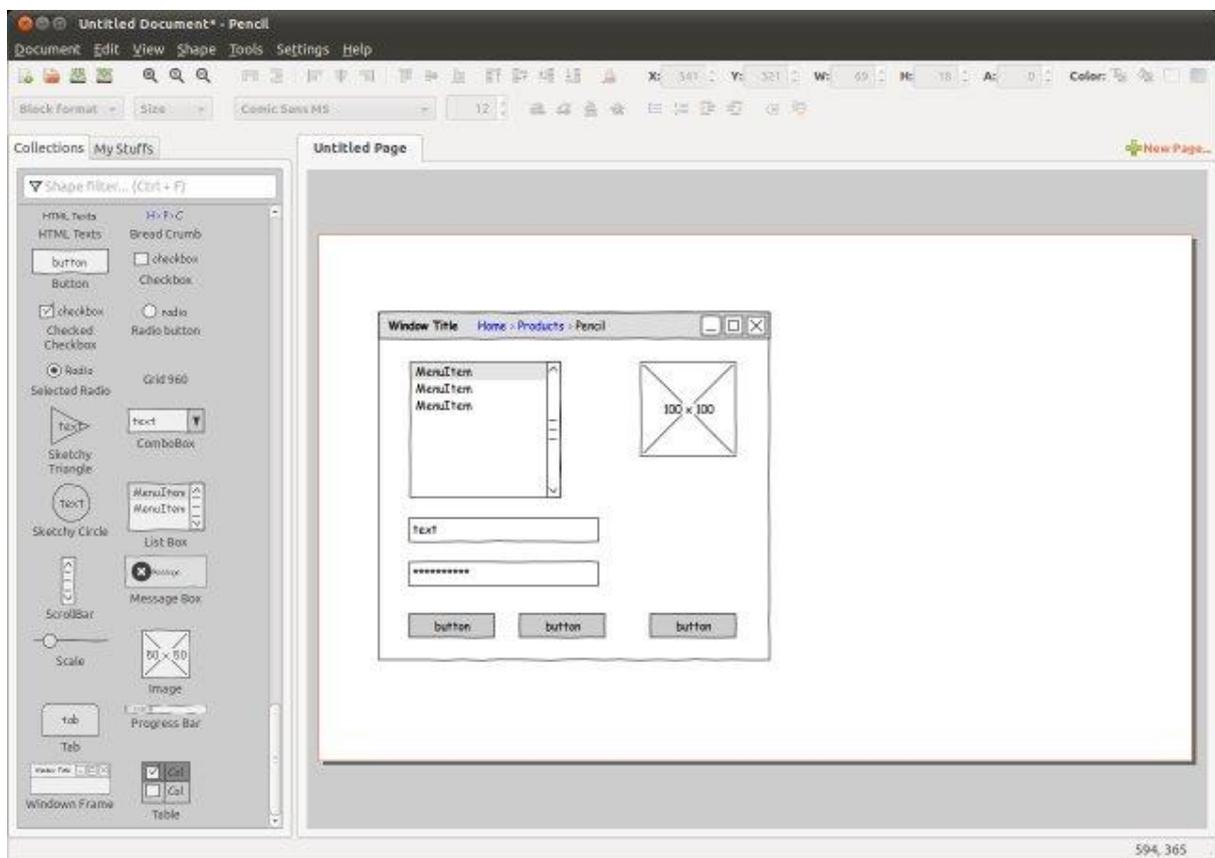


Figura 9. Diseño de pantalla en Pencil

Fuente: (Dedoimedo - Pencil, 2015)

Cuándo se le asigna un diseño a un programador de Front-End, este programador es el responsable de interpretar este diseño en HTML y CSS, utilizando lenguajes o las herramientas que él crea necesarias para replicar este diseño, usualmente se utiliza visual Basic.net y este programador según su propio criterio deduce que tamaños son conveniente, qué tamaño debe tener una tabla, un botón, aunque esto no es la manera más eficiente ya que dos programadores pueden tener criterios distintos, pueden darle un tamaño distinto a cada elemento, posicionarlo en distintas ubicaciones, asignarles un color distinto al mismo botón, en general el sistema podría ser completamente diferente a los diseños iniciales.

Se puede ejemplificar esta mala practica al crear un botón de guardar, el programador numero uno lo posiciona a la derecha con un tamaño de 15 pixeles de ancho y el programador numero dos lo posiciona a la izquierda con un tamaño de 40 pixeles, cuando el usuario final esté utilizando el sistema tendrán inconsistencias de qué en algunas pantallas o interfaces el mismo botón se encuentra en distintas posiciones y con distintos tamaños. Consecuencia de ello uno de los gestores de proyecto está encargado de revisar nuevamente las interfaces ya creadas en HTML y CSS verificar y rediseñar las interfaces si fuese necesario, este es un tiempo mal invertido, ya que no es necesario si se definieran funciones y estándares desde el inicio del desarrollo de un sistema.

Una vez teniendo el diseño el mismo programador desarrolla la lógica de Back-End y posterior que el sistema ya funciona según los requerimientos del mismo, éste se pasa a uno de los gestores el cual se encargará de las pruebas de el sistema con los usuarios finales o clientes.

Éste es un proceso en el cual el gestor se comunica con nuestros clientes para hacer pruebas del sistema sí no hubiese errores se firma un acta de satisfacción, pero si hubiese errores del sistema o no conformidades del usuario, estas se registrarán y se harán solicitudes de cambio a la sección de desarrollo nuevamente, estos cambios pueden ser a nivel de interfaz, de funcionamiento o solicitar una nueva funcionalidad o interfaz para este sistema.

4.2 SDLC con participación inicial de Front-End

El componente de Front-End es algo que ha venido evolucionando en los últimos años en los cuales se ha especificado distintas funciones, distintos roles distintos y procesos en los cuales esta involucrado principalmente el diseño y creación de interfaces, pero no solamente esas, también la experiencia que va a tener el usuario al utilizar el sistema, el proceso de creación de un sistema es decir, no hacer un proceso que sea tedioso para el usuario mientras están levantando los requerimientos o mientras se está desarrollando el sistema que esta sea una espera tan larga, que al recibir avances del desarrollo del sistema realmente sea lo que espera, a esto se le conoce como experiencia de usuario la cual es una de las funciones más novedosas que está teniendo el programador de Front-End.



Figura 10. Participación de Front-End en el SDLC.

Fuente: Propia

Las funciones principales de Front-End están comprendidas desde el análisis del sistema hasta la prueba del sistema una vez desarrollado. Entre estas etapas hay múltiples actividades con las cuales Front-End debe participar como ser el diseño del sistema y posteriormente el prototipo del sistema, la diferencia entre estos dos es que el diseño se refiere a una imagen de cómo se verá el sistema y el prototipo a como el sistema fluirá el mismo, que pasara cuando un usuario haga clic en un botón, para esto existe múltiples herramientas de las cuales unas de las principales es FIGMA.

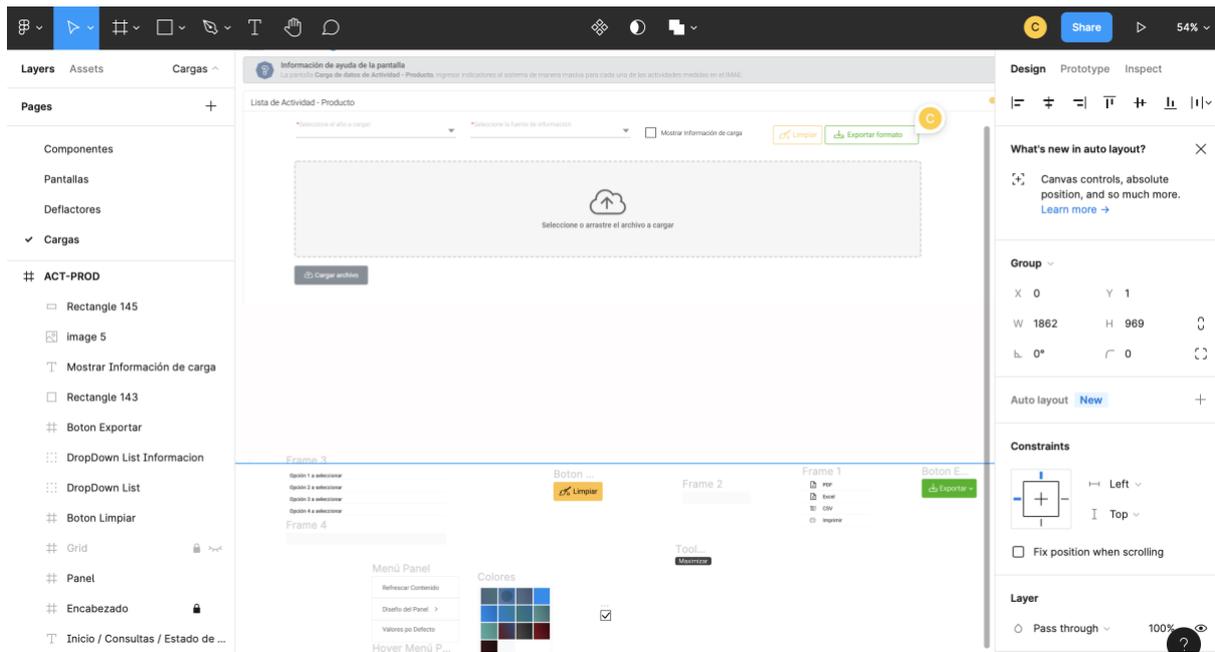


Figura 11. Diseño de pantalla en Figma
Fuente: Propia

4.3 Recolección de datos

Recolección de datos se realizó mediante dos instrumentos entrevistas semi estructuradas y consulta expertos, teniendo una participación total de 7 personas, 4 entrevistas a analistas de sistemas y desarrolladores y 3 participaciones de expertos.

4.3.1 Entrevistas

Las entrevistas semi estructuradas se realizaron con el propósito de identificar, describir y definir las actividades y funciones actuales y futuras del equipo de desarrollo de Front-End.

Partiendo de la experiencia técnica como empleados que buscan realizar actividades diarias eficientes con el objetivo general de culminar un ciclo de vida de software de manera exitosa, de esta manera obtuvimos datos que nos ayudaron a crear los descriptores de puestos y perfiles, agrupando las distintas actividades y funciones que los entrevistados describían, asimismo las etapas en las que estos roles podrían participar, describiendo los beneficios que

se podrían obtener de esta participación.

4.3.2 Consulta a expertos

La consulta de expertos se orientó a personal gerencial y administrativo como ser el líder técnico del proyecto y gestores de proyecto, estos tienen como objetivo gestionar aspectos técnicos y administrativos del proyecto como ser: tiempos, calidad, buenas prácticas, capacitaciones, entre otros.

Estos expertos describieron el tema desde un punto de vista gerencial, comentándonos los beneficios y recomendaciones de implementar la participación del equipo de desarrollo de Front-End, asimismo, experiencia previa de la implementación de una nueva tecnología la cual nos ayudó a generar la propuesta de implementación desde un punto técnico, administrativo y gerencial.

4.4 Análisis de los resultados

4.4.1 Proceso actual

Eficientar un proceso es un ciclo que nunca termina, ya que todas las instituciones buscan que sus procesos se desarrollen en el menor tiempo posible, que consuman la menor cantidad de recursos, ya sean materiales o humanos y al hablar específicamente del proceso del ciclo de vida del desarrollo de software según los datos recolectados se pudieron identificar algunos factores en los cuales los entrevistados asumía que estos son factores que reducen la eficiencia en el ciclo de vida del desarrollo de software principalmente los 4 siguientes:

- La comunicación
- Documentación ambigua
- Procesos con tiempo de respuesta muy extensa

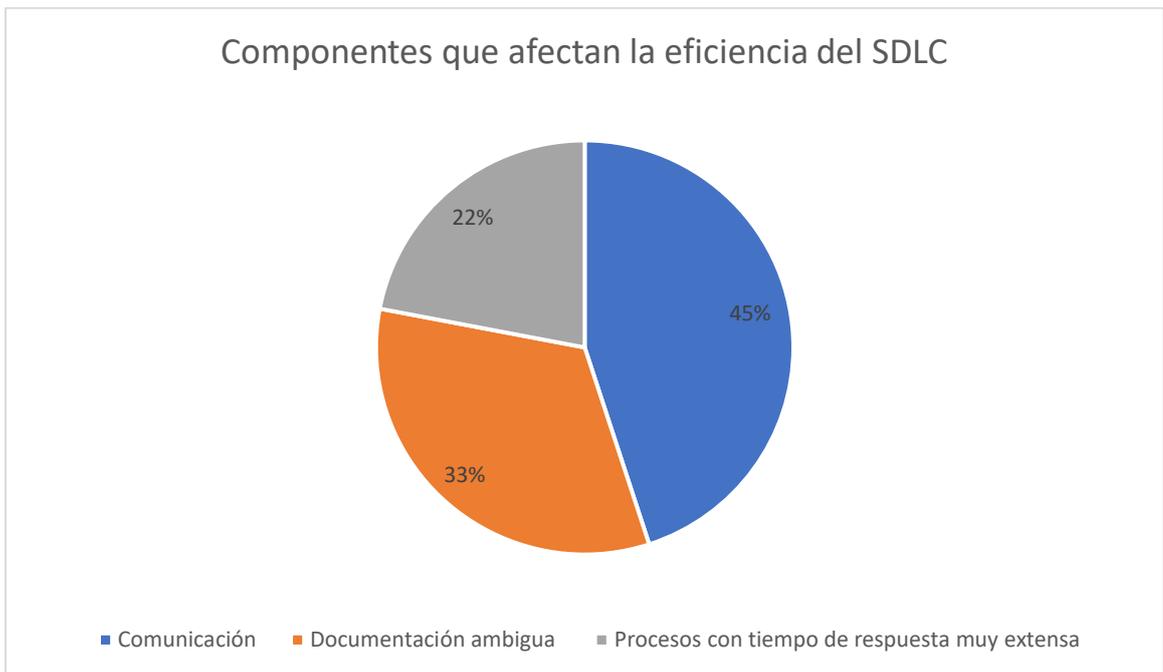


Figura 12. Componentes que afectan la eficiencia del SDLC.
Fuente: Propia

4.4.1.1 La comunicación

La comunicación es un factor muy importante en las relaciones de todo tipo, ya sean familiares, amorosas o laborales y en estas últimas principalmente enfocándonos en el desarrollo de software existen múltiples personas involucradas como ser el área de tecnología y el área del negocio que está interesada en un nuevo sistema pero existen pocas líneas de comunicación oficiales en las cuales se obtiene una respuesta rápida debido a la escalera de mandos, es decir el jefe del área del negocio le comunica al jefe del área de tecnología y éste a su vez a el gerente del proyecto, quien le comunica a su líder técnico, el cual le transmite el mensaje a un desarrollador y toda esta línea de comunicación es con el único objetivo de solicitar el cambio a nivel de gramática en un mensaje. Todo ese proceso antes descrito puede llegar a tardar 1-3 días.

4.4.1.2 Documentación ambigua

La documentación en un proyecto es el sustento legal en el cual se define lo que el usuario solicita y lo que el área de tecnología se compromete a entregar, pero es muy difícil cuando ninguna de estas dos partes se toma el tiempo de especificar cuál es eran las funcionalidades a las que se comprometen a entregar o a solicitar, ya que están acostumbrados a realizar solicitudes o gestiones de cambio, con la excusa de que son sistemas internos de que no generan costos ya que los desarrolladores y el personal de tecnología en su mayoría es personal de planta lo cual es un argumento completamente erróneo ya que aumentan el presupuesto por el pago de todo este personal causa retrasos en otros cronogramas y esto como resultado de una documentación poco detallada en la cual según sus estándares de desarrollo de sistemas existe y exige que se describan las entradas de información de un sistema y la salida que éste tendrá pero el usuario de Negocio por querer ahorrar tiempo en el proceso de documentación inicial no especifica todo el proceso que llevará llegar a esta salida.

4.4.1.3 Procesos con tiempo de respuesta muy extensa

Al ser el BCH una entidad gubernamental los procesos de autorización de documentación, contratación de servicios, solicitud de presupuestos entre otros procesos que incurren en el ciclo de vida del desarrollo de software pueden durar desde uno a siete años lo que conlleva que los usuarios finales o los solicitantes de un sistema presente quejas o no conformidad con los tiempos de espera.

4.4.2 SDLC con participación inicial de Front-End

Al escuchar a los entrevistadores, analizar los datos obtenidos de las entrevistas, indagar sobre las actividades y funciones que realiza un desarrollador de Front-End podemos identificar patrones en común entre estos tres aspectos. Partiendo de ello podemos segmentar estas funciones en las tres principales que se utilizan de manera internacional como en Colombia,

Estados Unidos, Suecia, Costa Rica, entre otros países estos tres roles son los siguientes:

- Diseñador
- Maquetador
- Programador lógico

Pero observamos que siempre mencionaban a un líder el cual distribuye a las actividades de Front-End, ayudaba y coordinaba que los componentes a desarrollar estuviesen listos en el tiempo asignado, indagaba sobre las buenas prácticas y actualmente está generando un estándar de diseño y mejoras de la experiencia de usuario en los sistemas.

Las actividades que los entrevistadores mencionaron fueron las siguientes:

- Participar en las reuniones de levantamiento de requerimientos.
- Diseñar pantallas.
- Realizar prototipos de pantallas.
- Programar la lógica de las pantallas.
- Replicar el prototipo de un sistema en HTML, CSS y JavaScript.
- Analizar el flujo de la aplicación.
- Realizar imágenes Docker de la aplicación.
- Indagar sobre nuevas tecnologías.
- Actualizar el Framework que se está utilizando.
- Comparativa entre herramientas de diseño.
- Implementar seguridad a nivel de Front-End.
- Crear paleta de colores para un sistema.
- Crear biblioteca de componentes genéricos en Angular.
- Crear biblioteca de componentes genéricos en Figma.
- Automatizar procesos para la implementación continua.

- Crear bibliotecas de clase de estilo.
- Confirmar entradas y salidas según requerimientos del sistema.

Al conversar con el gerente de proyectos no mencionaba que es muy difícil poder crear puesto de trabajo o roles tan específicos como recomiendan algunas instituciones debido a que la incorporación de estas tecnologías ha sido un proceso muy difícil con el cual se ha tenido que ir implementando poco a poco y que realizar una propuesta con funciones específicas para el equipo actual de desarrollo sería una gran ayuda y para ello se observó que todas las tareas y actividades que realiza el componente de Front-End se podrían agrupar de la siguiente manera:

Diseñador: es la persona que se encarga de la creación de diseños de pantallas utilizando herramientas como ser FIGMA, Adobe XD, entre otras, con el fin de definir tamaños de objetos en pixeles, como botones, tablas o listas, asimismo, definir una paleta de colores en formato RBG o hexadecimal.

Funciones:

- Diseñar pantallas.
- Realizar prototipos de pantallas.
- Analizar el flujo de la aplicación.
- Crear paleta de colores para un sistema.
- Crear biblioteca de componentes genéricos en Figma.
- Comparativa entre herramientas de diseño.

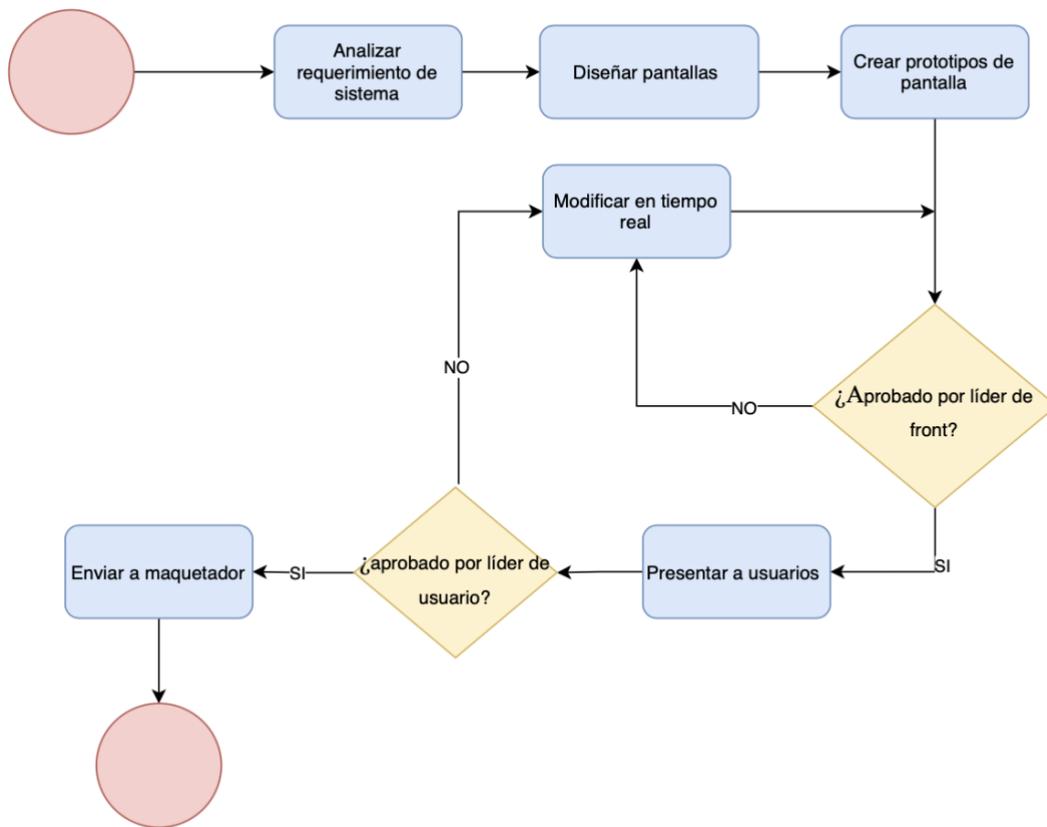


Figura 13. Flujo de actividades del diseñador.

Fuente: Propia.

Etapas en las que el diseñador tiene una participación importante:

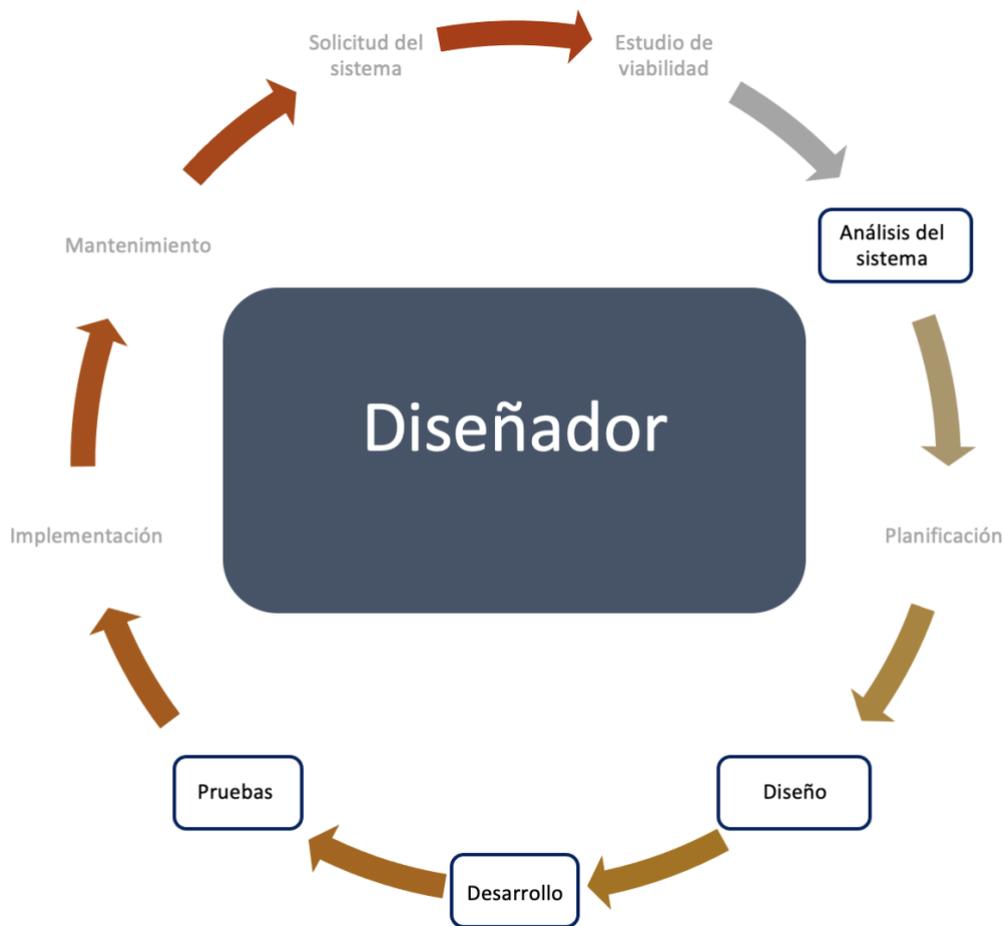


Figura 14. Etapas del SDLC en las que el diseñador participa.
Fuente: Propia.

Se identificó que las etapas del ciclo de vida del desarrollo de software en las cuales participará el rol del diseñador son: análisis del sistema, diseño, desarrollo y pruebas.

En la etapa de análisis de sistemas el diseñador tiene comunicación con el usuario Negocio y el gestor del proyecto ya que se diseñan en tiempo real y en colaboración con las partes antes mencionadas las pantallas mediante la herramienta de firma lo que ayuda a crear diseños y prototipos más acertados a las necesidades del usuario.

En la etapa del diseño del sistema el diseñador afina los tamaños, colores, posiciones

según los estándares creados para el sistema en la misma etapa. En la etapa de desarrollo el diseñador asiste al maquillador y el programador lógico y valida que el diseño cumpla el flujo y la apariencia especificada en los prototipos y diseños entregados.

La etapa de pruebas es una etapa que está en constante participación con el diseñador ya que no es necesario que un producto esté completamente terminado para que llegue a esta etapa, es decir al terminar solamente el diseño solamente el diseño pasará a una revisión con el líder técnico, posteriormente con el usuario de Negocio para realizar ajustes en etapas tempranas.

Maquetador: es la persona que se encarga de interpretar los diseños y prototipos en lenguaje HTML, CSS, SASS y JavaScript, realizando la apariencia de un sistema responsiva para múltiples dispositivos.

Algunos de los entrevistados recomiendan que el rol del diseñador y el maquetador puede ser realizado por la misma persona, esto solamente cuando la cantidad de personal es limitada.

Funciones:

- Crear biblioteca de componentes genéricos en Angular.
- Crear bibliotecas de clase de estilo.
- Replicar el prototipo de un sistema en HTML, CSS y JavaScript.

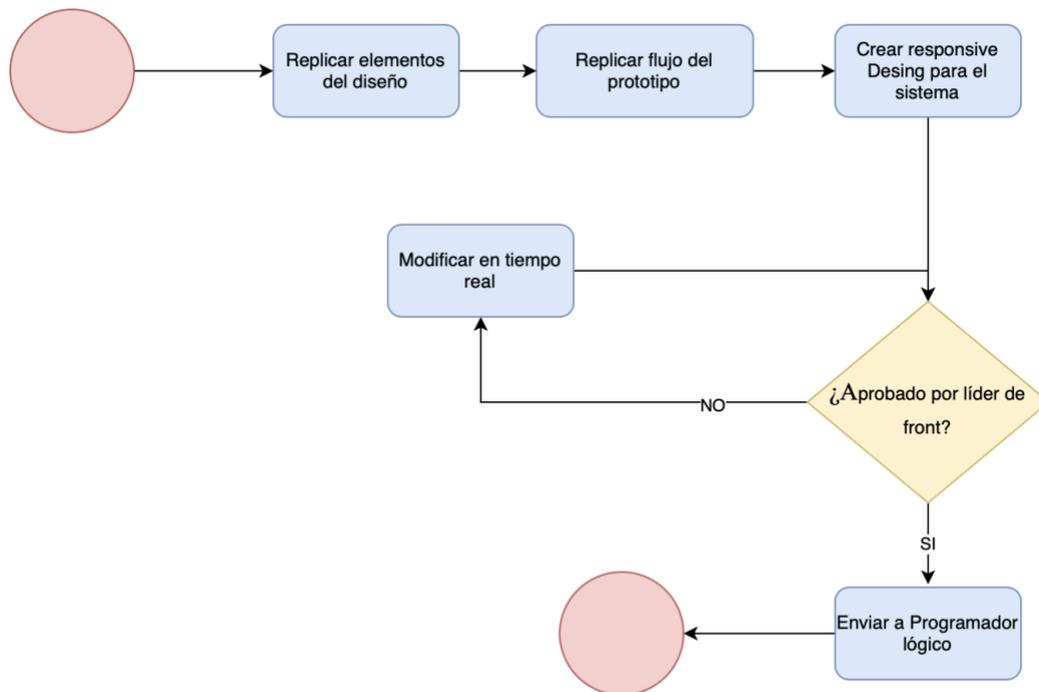


Figura 15. Flujo de actividades del maquetador.
Fuente: Propia.

Etapas en las que el maquetador tiene una participación importante:

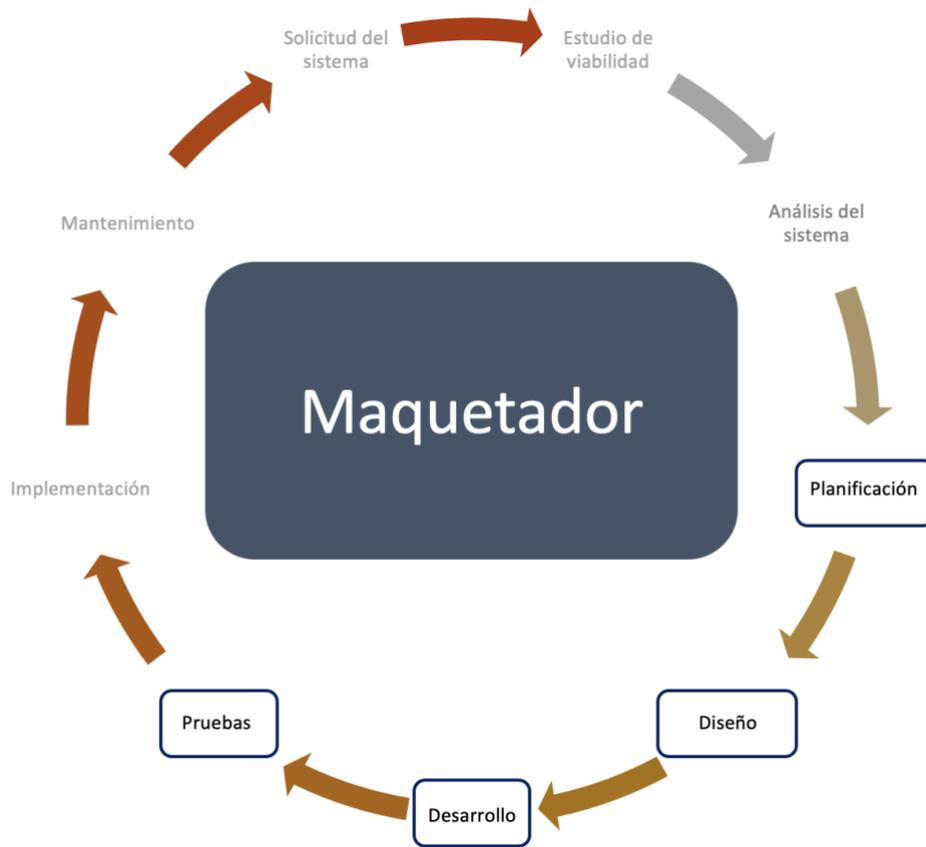


Figura 16. Etapas del SDLC en las que el maquetador participa.
Fuente: Propia.

Se identificó que las etapas del ciclo de vida del desarrollo de software en las cuales participará el rol de maquetador son: planificación, diseño, desarrollo y pruebas.

En la etapa de planificación el maquetador analizará todos los diseños y prototipos que fueron entregados por el diseñador, para dar una estimación de tiempos según su criterio de expertos, lo cual ayudará a crear un cronograma de trabajo y un esquema de entregables para crear segmento de trabajo con la metodología de Scrum.

En la etapa de diseño el diseñador y el maquetador trabajarán de la mano para modificar flujos o agregar elementos a los diseños previamente creados esto ya que hay flujos que son

demasiado tratados en la maquetación.

En etapa de desarrollo se replicará mediante HTML, CSS y Java script todos los diseños y prototipos de interfaces que acordaron con el diseñador en la etapa de diseño.

En la etapa de pruebas el maquetador analizará todos los ajustes y comentarios solicitados por los usuarios.

Programador lógico: es el que se encarga del diseño y la interpretación de la estructura de las API REST y de esta manera poder mostrar los datos en el formato o la apariencia que el usuario solicitó y también poder estructurar los datos que el usuario ingrese de una manera que se puedan almacenar en la base de datos por medio de un intermediario como hacer una API Rest.

Funciones:

- Participar en las reuniones de levantamiento de requerimientos.
- Programar la lógica de las pantallas.
- Realizar imágenes Docker de la aplicación.
- Indagar sobre nuevas tecnologías.
- Actualizar el Framework que se está utilizando.
- Implementar seguridad a nivel de Front-End.
- Automatizar procesos para la implementación continua.
- Confirmar entradas y salidas según requerimientos del sistema.

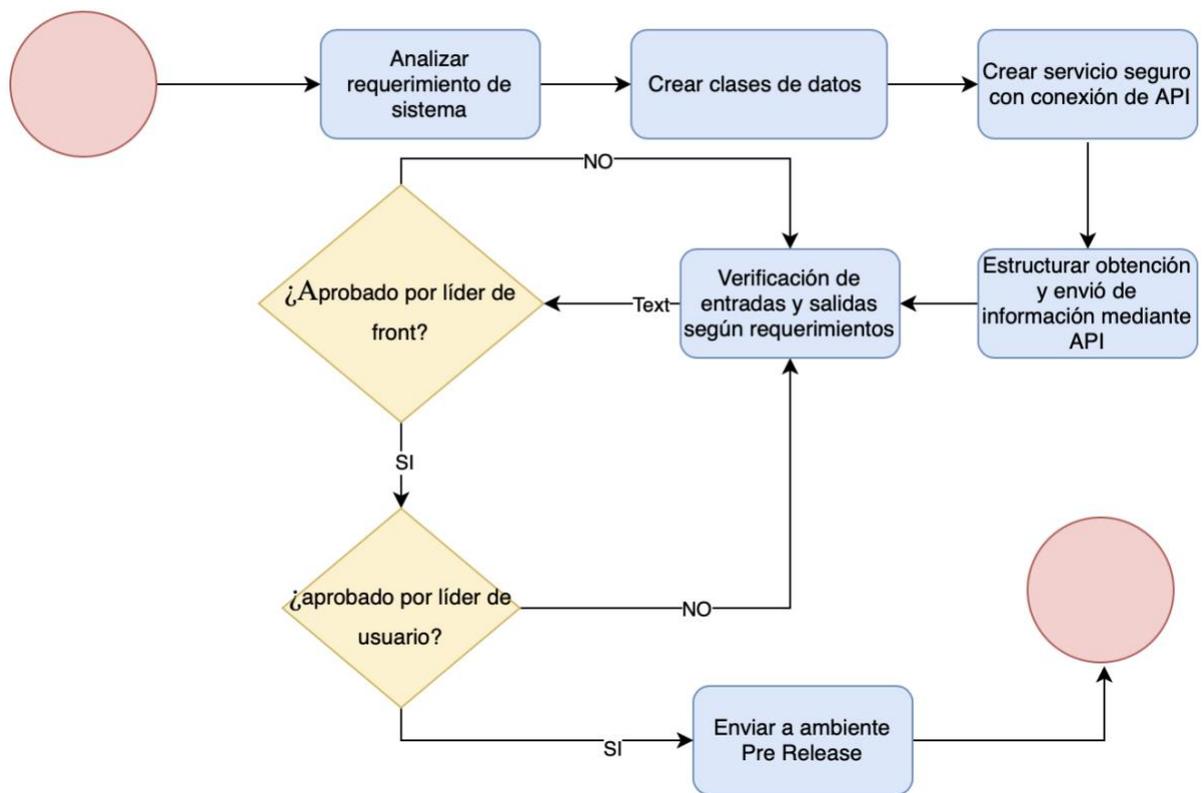


Figura 17. Flujo de actividades de el programador lógico.
 Fuente: Propia.

Etapas en las que el programador lógico tiene una participación importante:

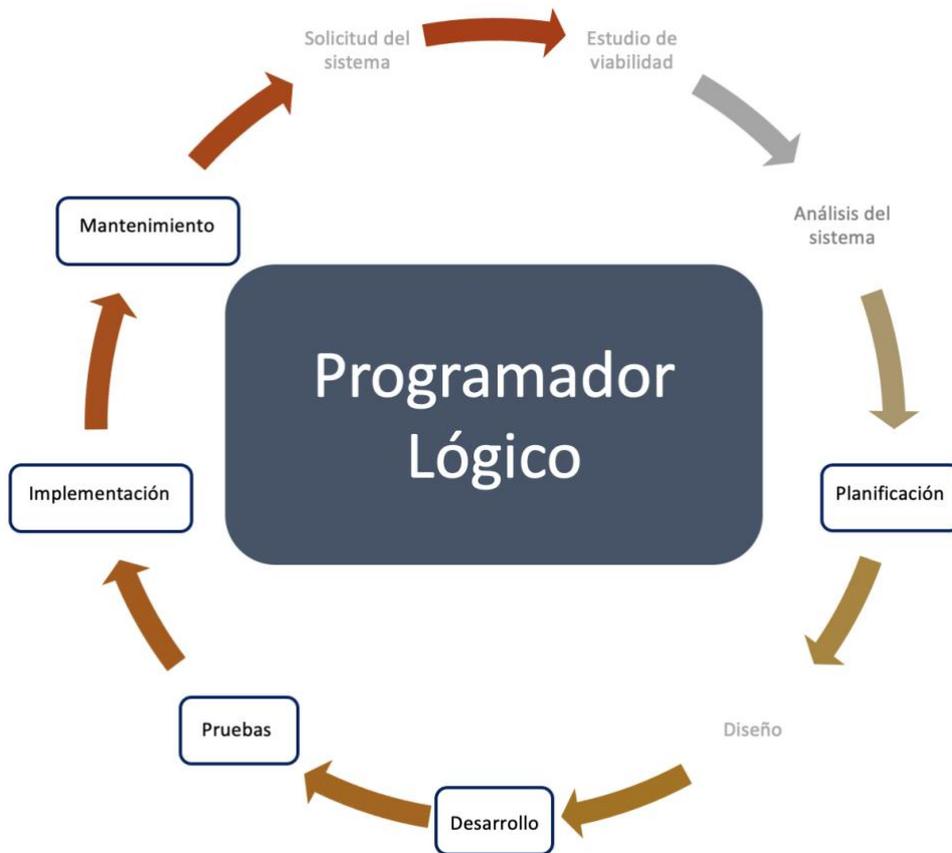


Figura 18. Etapas del SDLC en las que el programador lógico participa.

Fuente: Propia.

Se identificó que las etapas del ciclo de vida de software en las cuales participará el rol de desarrollador lógico son planificación, desarrollo, pruebas, implementación y mantenimiento.

En la etapa de planificación el programador lógico analizará las entradas y salidas de información, estructuras de datos, validación de información y seguridad de la misma, con el fin de poder dar estimado de tiempo, cantidad de actividades y entregables para poder realizar un cronograma de actividades más detallado y preciso.

En la etapa de desarrollo utilizando distintos Framework definidos al inicio del ciclo de vida

del desarrollo de software, el programador lógico creará las estructuras de datos y todo lo necesario para poder mostrar los datos como son solicitados en los requerimientos del sistema es decir mostrar las salidas y obtener las entradas de información según los requerimientos del usuario.

En la etapa implementación el programador lógico una vez que tenga la aprobación del usuario y del líder técnico creará imágenes Docker y serán publicadas en un ambiente de pre release.

En la etapa de mantenimiento reprogramó lógico del soporte temprano y colaborará con el departamento de mantenimiento de sistemas.

Analizar las respuestas de las personas entrevistadas e identificar las actividades que realiza el personal de Front-End, enseguida agrupar estas actividades, identificar en las etapas del ciclo de vida del desarrollo de software que puedan participar, fueron el inicio para detectar en qué mejoraría el ciclo de vida del desarrollo de software al implementar estos roles estas agrupaciones de actividades en el personal de desarrollo y hemos descubierto algo que los entrevistados expertos en el área nos mencionaba y más que aumentar la eficiencia en grandes escalas de tiempo es agregar un valor agregado al ciclo de vida de desarrollo de software interno en el BCH es decir las reuniones con los usuarios para el levantamiento de requerimientos siempre han existido, Pero si participaste el personal de Front- End en estas reuniones agregamos diseños de pantallas muy parecidos a lo que será el producto final y aún más se presentan prototipos que que reflejarán el aspecto final del sistema. Esto solamente es la intervención del diseñador posteriormente en el diseño de la aplicación el maquetador podrá entregar de manera más rápida los productos o entregas programadas, aunque aún no arrojen información de salida o almacene los datos de entrada, pero se podrá ver de manera temprana el resultado final a nivel de diseño, flujo y apariencia. Y por último el programador lógico

disminuirá los tiempos de programación de crear estructuras de datos ya que han tenido una continua comunicación con los desarrolladores del equipo de imagen lo que disminuye la modificación y ajustes en las pantallas, lo que con lleva a un menor tiempo de desarrollo de las aplicaciones.

Así también hemos podido identificar según los relatos que una persona especializada en un aria de Mora una cantidad de tiempo menor a un Full Stack para ello podemos verlo en la siguiente tabla:

Tabla 6. Comparación de tiempos rol tradicional vs distintos roles de Front-End

Componente	Diseñador	Maquetador	Programador Lógico	Full Stack
Grid	10 min	15 min	20 min	50 min
Lista	5 min	10 min	20 min	50 min
Boton	5 min	10 min	20 min	50 min
DDL	5 min	10 min	20 min	50 min
Estructura de datos	-	-	20 min	50 min
Componente de carga	-	-	20 min	50 min

Estos tiempos son establecidos según los entrevistados a los cuales se les mostraron las siguientes imágenes de componentes, con el fin de que establecieran tiempos aproximados según su criterio de experto:

Grid

No.	Codificación	Descripción	Nivel
1	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
2	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
3	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
4	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
5	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
6	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
7	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
8	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
9	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ
10	Texto ABC...XYZ	Texto ABC...XYZ	Texto ABC...XYZ

1 de 833 páginas (8322 registros)

Figura 19. Grid con funciones básicas
Fuente: Propia.

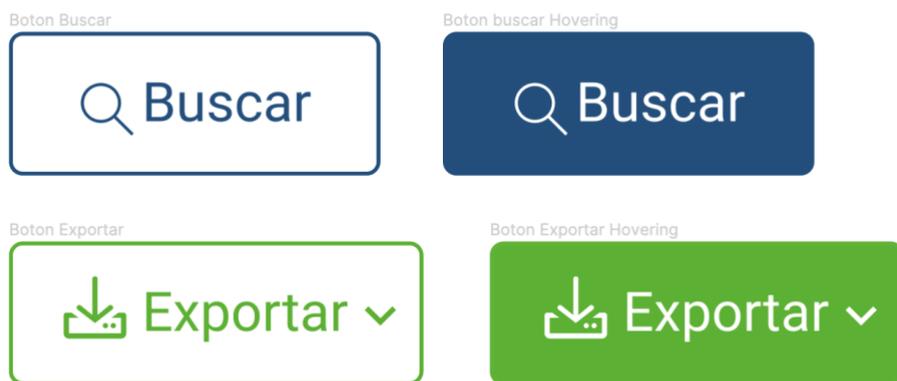


Figura 20. Botones y diseño con hover
Fuente: Propia.

*Seleccione el tipo de pantalla de carga:

Opciones de DDL

- Opción 1 a seleccionar
- Opción 2 a seleccionar
- Opción 3 a seleccionar
- Opción 4 a seleccionar

Figura 21. Drop Down List con funcionalidad básica
Fuente: Propia.

Text Box Icon

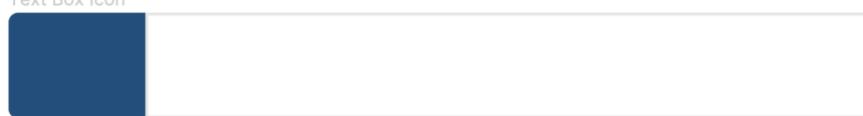


Figura 22. Input con espacio para un icono.
Fuente: Propia.

Generar una nueva equivalencia

Boto... Boto...

Guardar Cancelar

Figura 23. Modal vacío con botones inferiores.
Fuente: Propia.

4.4.3 Resultados según variables

Tabla 7. Resultados según variables

Variables	Resultado
Personal humano	<p>El personal humano puede ser una nueva contratación o reasignación de personal actual según en sus habilidades y previas o futuras áreas capacitadas. Esto mejorará la calidad de los productos finales, la motivación y aumenta los indicadores de personal capacitado según áreas de trabajo.</p> <p>Se identificó que las responsabilidades se distribuyen en todo el equipo y no solo se depende del criterio y presencia de un desarrollador Full-Stack.</p>
Clima motivacional	<p>La comunicación es un factor fundamental en el SDLC, específicamente en el área del BCH, producto a procesos tediosos de comunicación, el crear canales y utilizar herramientas de comunicación, creación de requerimientos y diseños en tiempo real, aumentará la comunicación y participación de mas miembros en el SDLC.</p>
Carga laboral	<p>La carga laboral aumenta en cantidad de asignaciones para el equipo de Front-End, pero según el criterio de los expertos entrevistados disminuirá las solicitudes de gestión de cambio, lo que en consecuencia disminuye la cantidad de horas invertidas en el SDLC.</p> <p>Lo que interpretamos que al tener un personal especializado el tiempo para realizar las actividades será menor y con un nivel de calidad mucho mayor.</p>

CAPÍTULO V. CONCLUSIONES Y RECOMENTACIONES

5.1 Conclusiones

- Los usuarios de negocio y el equipo de tecnología son personas con poca resistencia al cambio, al utilizar nuevas herramientas, aumentado la calidad de la apariencia de los sistemas, son aspectos que motivan aún más la creación de nuevos sistemas, sistemas que tienen una usabilidad mayor y un menor porcentaje menos de quedar en desuso. Ya que el producto final fue construido con la participación de ambas partes.
- En las instituciones gubernamentales en Honduras la creación de un sistema es un proceso que puede tardar desde seis meses hasta un año al realizar el ciclo de vida del desarrollo de software completo, ya que cada etapa conlleva distintos procesos muy tardados por aspectos de aprobación, líneas de comunicación, algún tipo de licitación entre otros factores, así que el departamento de tecnología participa en una metodología de mejora continua para que estos procesos sean cada vez más eficientes y una de las etapas en las que más enfocan es poder tener entregas más certeras con los usuarios es decir entregar etapas o funcionalidades de un sistema y que el usuario final las apruebe con la menor cantidad de ajustes posibles. El rol de Front-End está jugando un papel muy importante en la reducción de ajustes, ya que el usuario final no observa la lógica del sistema sólo analiza las entradas o salidas de información que se muestran en las interfaces del sistema y es donde Front-End mejora estos procesos.
- El ciclo de vida de desarrollo de un sistema utilizado actualmente por el BCH en la sección de desarrollo se adapta a las necesidades y procedimientos de esta sección, aunque este ciclo de vida está en continuo proceso de mejora con la finalidad de

ofrecer un servicio de mayor calidad.

- El equipo de desarrollo de Front-End está liderado por un gestor de proyecto el cual ha cultivado la adaptación a nuevas tecnologías, cambios y mejoras en proceso. Es muy interesante poder observar que todos los integrantes de esta sección están dispuestos a realizar e implementar cualquier cambio ya sea de tecnología, infraestructura, organización o cualquier otro con el objetivo de estar innovando y mejorando la calidad de sus servicios.
- La metodología en las instituciones de contar con programadores Full Stack está evolucionando a tener desarrolladores como un conocimiento, es decir un conocimiento muy amplio en todo el ciclo de vida del desarrollo de software, pero especializados en un área ya sea Front, Back, base de datos, arquitectura o infraestructuras que obtienen una mayor calidad en sus trabajos.
- La falta de un estándar en el diseño de un sistema provoca un bajo nivel de usabilidad, y es más común observar en una metodología tradicional donde un desarrollador Full Stack realiza todas las actividades en base a su criterio, es decir ubica los componentes de una pantalla donde le sea más conveniente, pero no manejan tamaños específicos, posiciones, colores, textos y esto provoca confusión en los usuarios a diferencia de implementar el rol del diseñador experto en experiencia de usuario el cual al iniciar un sistema crea un estándar de paletas de colores, componentes genéricos, posiciones en las pantallas, tamaños en pixeles, textos entre otros.
- La experiencia de usuario es un concepto y área de conocimiento desconocido para mucho en el área de la tecnología, por ejemplo algunos gestores de proyecto definen

la experiencia de usuario como estrategias de marketing para un sistema y otros no conocen el concepto de experiencia de usuario, el componente de Front-End está en evolución desde el año 2014 al 2016 y actualmente en los países de Latinoamérica se está implementando nuevas tecnologías y esta implementación en las entidades gubernamentales tiene un proceso mucho, pero el BCH está en proceso de capacitación del personal para aumentar la calidad de sus sistemas e infraestructura.

5.2 Recomendaciones

- Esta investigación cualitativa se puede usar de base para iniciar una investigación cuantitativa y poder formular y comprobar una hipótesis que incluir al equipo de Front-End desde el inicio del ciclo de vida del desarrollo de software y basado en las buenas prácticas de ITIL versión cuatro, reduce la cantidad de horas empleadas en el mismo y consecuencia de esto aumenta la satisfacción en los usuarios. Creando roles y a su vez tareas específicas en cada etapa del ciclo de vida del desarrollo de sistemas.
- Se recomienda crear perfiles y descriptores de puestos en el área de tecnología, específicamente en el área de desarrollo de sistemas, estos puestos se deben definir en base a los roles que se estarán implementando en el BCH basados en estándares internacionales recomendaciones de organizaciones especializadas en Front-End.
- Se recomienda la integración de múltiples herramientas que permiten una fluida comunicación entre el equipo desarrollo y los usuarios de negocio herramientas como Figma para el diseño en tiempo real, Azure DevOps para la documentación digital de los requerimientos de usuario, seguimiento de tareas y la implementación y despliegue continuo.
- Se recomienda capacitar al personal en áreas específicas como ser el diseño y la experiencia de usuario, HTML, CSS y JavaScript avanzado, Azure DevOps, TypeScript, angular y React.
- Se recomienda la creación de estándares de diseño previo al desarrollo de cada sistema, en este estándar se crearon la paleta de colores que se utilizarán en cada uno de los componentes implementados en el sistema, biblioteca de clase a utilizar, tamaño de textos, tipo de alertas, notificaciones, mensajes de errores y estándares de buenas prácticas de programación.

- Se recomienda crear una biblioteca de componentes genéricos desarrollados en HTML y CSS con el fin de no depender de ninguna librería de terceros, crear sus propias tablas o grid, listas, botones, ayudas, entre otros componentes que sean implementados continuamente.
- Se recomienda crear reuniones en las cuales participen todos los equipos de desarrollo con el fin de tener una mejor comunicación crear procesos adaptados a la institución canales de comunicación oficiales los cuales pueden ser correo institucional o discusiones mediante Azure DevOps, Slack u otra herramienta de respuesta en tiempo real.
- Se recomienda utilizar la propuesta de implementación presentada en esta investigación de tesis con el objetivo de definir los roles del equipo de Front-End, mejorar la satisfacción del usuario de negocio y eficientar los tiempos en las etapas del SDLC.

CAPÍTULO VI. PROPUESTA

6.1 Definición y uso de roles en el equipo de Front-End

6.2 Justificación

La creación de sistemas estadísticos web es cada vez mayor, por tal razón es necesario eficientar los tiempos en el ciclo de vida del desarrollo de software y mejorar la calidad de los sistemas desarrollados

La solicitud de nuevos sistemas estadísticos web ha aumentado en los últimos años, pero el ciclo de vida de nuevos desarrollos de software dentro del BCH es tiene tiempos muy largos, por lo cual es necesario eficientar los tiempos en cada una de las etapas de este ciclo.

Al realizar esta investigación con una metodología cualitativa y un alcance descriptivo, se escribieron los procesos, funciones y etapas en las cuales el personal actual de Front-End tiene intervención y se realizó una investigación de ofertas de empleo a nivel nacional e internacional para el desarrollo del componente de Front-End descubriendo puestos como ser diseñadores, maquetadores, programadores, tester, entre otros, posteriormente se agruparon las actividades que realizan los integrantes del equipo de Front-End y se agruparon según los puestos y ofertas laborales previamente mencionadas obteniendo una agrupación final en tres roles principales los cuales son: diseñador, maquetador y programador lógico.

6.3 Alcance

6.3.1 Objetivo general

- Implementar el rol de diseñador, maquilador y programador lógico en el equipo de desarrollo de Front-End el del BCH.

6.3.2 Objetivos específicos

- Capacitar al personal del área de desarrollo en temas relacionados con la experiencia de usuario, diseño de interfaces, angular, Docker y Scrum.
- Organizar a los integrantes del equipo de Front-End según sus habilidades acorde a los roles de diseñador, agitador y programador lógico.
- Definir según los lineamientos y recomendaciones del BCH, los descriptores y perfiles de puestos para los roles de diseñador, maquilador y programador lógico.

6.4 Descripción y desarrollo

6.4.1 Descripción

En esta sección se describe la propuesta para eficientar los tiempos en las en algunas de las etapas del ciclo de vida del desarrollo de software en la sección de proyectos de banca central en el banco central de Honduras y implementando la participación del equipo de Front-End definiendo roles y sus funciones para cada una de las etapas del ciclo de vida de desarrollo de software.

Recientemente se implementó una nueva arquitectura de desarrollo en la cual se divide arquitectura del lado del servidor el cual es desarrollado por el equipo de Back-End y el lado del cliente el cual es desarrollado por el equipo de Front-End, pero la participación de este último en el ciclo de vida del desarrollo de software se limita a la etapa de desarrollo.

El equipo de Front-End replica diseños realizados por usuarios de negocio, gestores de proyecto, enlace del proyecto u otras personas que intervienen en la etapa del análisis del sistema, producto de ello las funciones y actividades que realiza el equipo de Front-End están limitados al desarrollo de las aplicaciones como tal pero fábricas de software internacionales incluyen a personal de este equipo en los análisis del sistema y luego de analizar esta estructura en empresas internacionales, oferta de empleo en otros países, en algunas empresas nacionales y después de analizar los resultados provenientes de las encuestas y consultas a expertos, se propone crear tres roles que sean desempeñadas por los integrantes del equipo de Front-End del área de desarrollo del Banco Central de Honduras realizando una prueba piloto previo a

contratar nuevo personal que como requisito mínimo se solicita que estas personas cuenten con un título universitario de ingeniería en sistemas o carreras a fines al área de tecnología, esto por directrices y procesos internos del departamento de Gestión de Talento Humano, en la cual se distribuyen las actividades de cada uno de estos roles en el personal permanente de esta institución, los roles que se implementarían serán:

- Diseñador
- Maquetador
- Programador lógico

De manera inicial se analizarán cada una de las habilidades y preferencias del personal de planta de esta sección para realizar una agrupación y asignación de estos roles según las habilidades y las preferencias que tenga este personal, de esta manera se podrá enfocar capacitaciones específicas de áreas específicas para un rol en específico, es decir el personal que tenga habilidades de dibujo, diseño, conocimiento de herramientas o plataformas de diseño se les dará una capacitación en áreas de experiencia de usuario, interfaz de usuario, herramienta como firma y se les asignará el rol de diseñador para la prueba piloto.

Luego de definir y asignar esos roles se comenzará a implementar la participación de el equipo de Front-End en las distintas etapas del ciclo de vida del desarrollo de software, en la prueba piloto no se realizará un sistema completo, sino que se realizará en El desarrollo de un módulo del sistema que se está desarrollando en ese momento, de esta manera participará en el análisis y discusión de los requerimientos

del sistema con la usuario de negocio, la planificación, el diseño, y subsecuentemente las etapas que siguen hasta culminar el ciclo de vida del desarrollo de software de un módulo en específico.

Al combinar el ciclo de vida desarrollo de software de un módulo se medirá en tiempos de desarrollo tiempos para finalizar cada una de las etapas, satisfacción de los usuarios de negocio, experiencias del desarrollador e integrante del equipo de Front-End con el fin de mí medir el alcance, las ventajas y desventajas que tuvo esta prueba piloto y poder concluir si es necesario utilizar un mayor presupuesto para la contratación de nuevo personal y asignarlo a estos nuevos roles.

6.4.2 Desarrollo

En el área de tecnología tratar de implementar una propuesta que involucre la contratación de nuevo personal, adquisición de nuevo equipo, implementación de nuevas estructuras, o cualquier otro objetivo que involucre mayor presupuesto es una tarea muy difícil debido a que frecuentemente el área de tecnología se considera un gasto en una inversión, para ello previo a solicitar la contratación de nuevo personal para poder cubrir los tres roles propuestos para el equipo de desarrollo de Front-End se realizará una prueba piloto la cual realice todo el ciclo de vida del desarrollo de software en un módulo aleatorio y poder incluir al equipo de Front-End definiendo sus funciones y utilizando según su rol los descriptores y perfiles de puesto siguientes:

DISEÑADOR

DESCRIPCIÓN Y PERFIL DEL CARGO

2022

Título del cargo	Área a la que pertenece
Diseñador	DPT
Cargo al que reporta	Sub-área a la que pertenece
Líder de Front-End	SPBC
Cargos que le reportan	Fecha de elaboración
N/A	20/06/2022

Misión

El diseñador de encargará de analizar e interpretar los requerimientos de usuario, creando diseños en colaboración con el usuario de negocio y el área de desarrollo aplicando conceptos de UI/UX los cuales ayudarán a producir un aumento satisfacción del cliente al recibir productos de mejor calidad.

Requisitos administrativos

- Edad: en un rango de 21 a 35 años
- Sexo: Indistinto
- Domicilio: Tegucigalpa, FM

Condiciones sociales

Tabla 8. Condiciones sociales del diseñador

Condiciones	Frecuentemente	Ocasional
Atiende Proveedores		X
Atiende Clientes	X	
Atiende Empleados	X	
Comparte sitio de trabajo		X
Requiere viajar		X
Jornada de trabajo		

Competencias Organizacionales

Tabla 9. Competencias Organizacionales

Competencia	Grado
Orientación a la calidad	B
Orientación al servicio	A
Comunicación	A

Competencias Función/Técnicas

Tabla 10. Competencias Función/Técnicas

Competencia	Grado
Comunicación eficaz	B
Ética	A
Auto aprendizaje	A
Innovación	A

Grados de Competencia:

A: Avanzado B: Experto C: Intermedio D: Básico

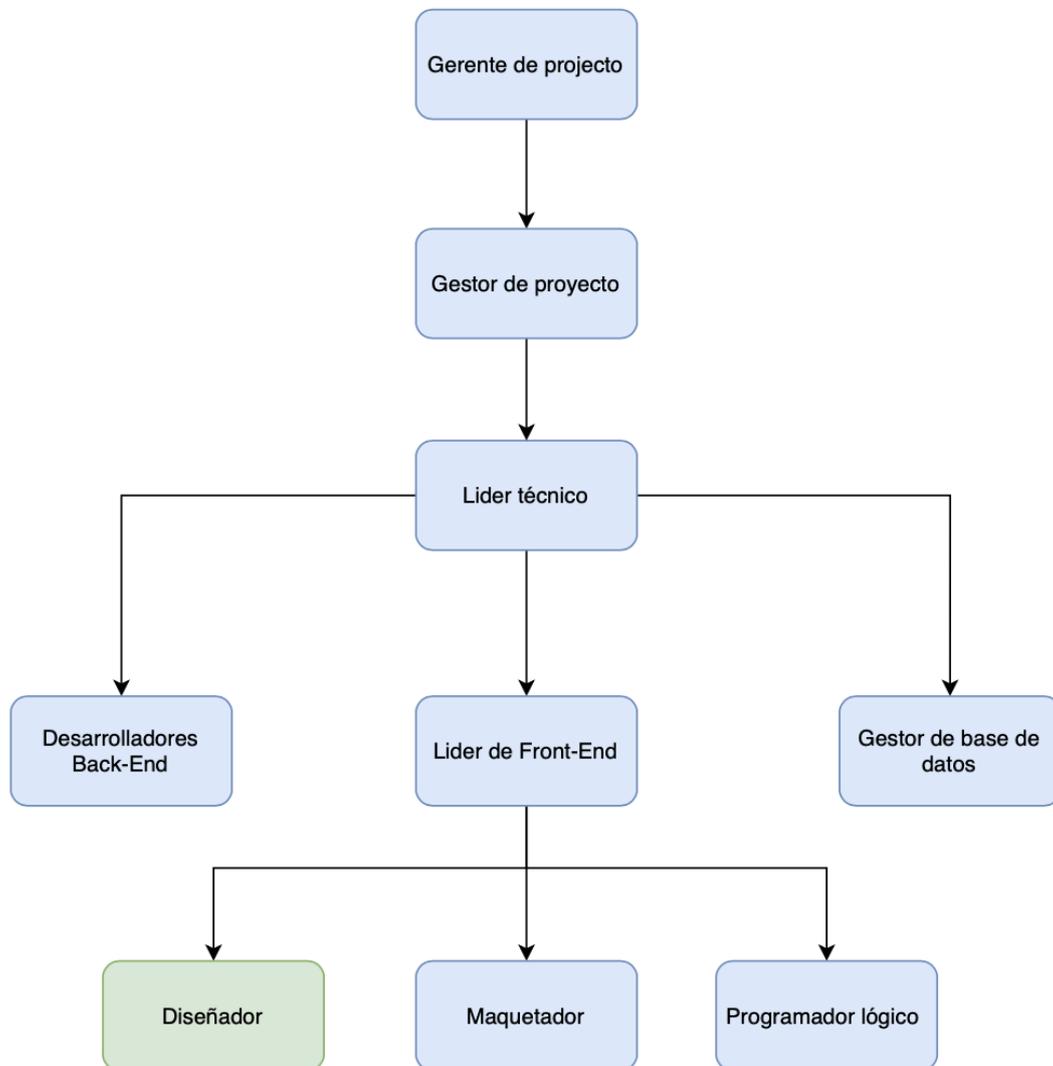
Conocimientos	
Estudios:	
Ingeniería en sistemas, licenciatura en Diseño de Comunicación/Diseño Gráfico o carreras a fines.	
Áreas de experiencia:	
<ul style="list-style-type: none">• UI/UX• HTML• CSS• JavaScript• Photoshop• Ilustrador• Figma• Abode XD• Metodologías ágiles• Azure DevOps	

Tabla 11. Responsabilidades del diseñador

Responsabilidades	Periodicidad			
	Diaria	Semanal	Mensual	Ocasional
Diseñar pantallas	X			
Realizar prototipos de pantallas	X			
Analizar el flujo de la aplicación	X			
Crear paleta de colores para un sistema				X
Crear biblioteca de componentes genéricos en Figma				X
Trabajar con los equipos de desarrollo para asegurar que los diseños sean debidamente probados e implementados	X			
Entregar controles UI/UX				X
Trabajar eficazmente en varios proyectos al mismo tiempo	X			

Reunir y evaluar los requisitos de los usuarios en colaboración con gestores y usuarios de negocio.			X	
Preparar y presentar borradores a los equipos internos y a las principales partes interesadas		X		
Identificar y solucionar problemas de UI/UX (por ejemplo, capacidad de respuesta, diseño)	X			
Realizar ajustes de diseño basados en los comentarios de los usuarios	X			
Realizar ajustes de diseño basados en los comentarios del líder técnico y gestores.	X			
Participar en capacitaciones y reuniones.			X	
Cumplir con las normas de estilo en cuanto a fuentes, colores, diseño e imágenes.	X			

Organigrama



Maquetador Web

DESCRIPCIÓN Y PERFIL DEL CARGO

2022

Título del cargo	Área a la que pertenece
Maquetador Web	DPT
Cargo al que reporta	Sub-área a la que pertenece
Líder de Front-End	SPBC
Cargos que le reportan	Fecha de elaboración
N/A	20/06/2022

Descripción

El maquetador web será el responsable de crear el diseño de las aplicaciones web en HTML, CSS y JavaScript, haciendo uso de Frameworks como hacer: angular, Vue o React. Basándose en los diseños y prototipos entregados por el diseñador, asimismo deberá apegarse a los estándares, posiciones, tamaños y paleta de colores definidos previamente.

Requisitos administrativos

- Edad: en un rango de 21 a 35 años
- Sexo: Indistinto
- Domicilio: Tegucigalpa, FM

Condiciones sociales

Tabla 12. Condiciones sociales del maquetaador

Condiciones	Frecuentemente	Ocasional
Atiende Proveedores		X
Atiende Clientes	X	
Atiende Empleados	X	
Comparte sitio de trabajo		X
Requiere viajar		X
Jornada de trabajo		

Competencias Organizacionales

Tabla 13. Competencias Organizacionales del maquetaador

Competencia	Grado
Orientación a la calidad	B
Orientación al servicio	A
Comunicación	A

Competencias Función/Técnicas

Tabla 14. Competencias Función/Técnicas del maquetaador

Competencia	Grado
Comunicación eficaz	B
Ética	A
Auto aprendizaje	A
Innovación	A

Grados de Competencia:

A: Avanzado **B:** Experto **C:** Intermedio **D:** Básico

Conocimientos

Estudios:

Ingeniería en sistemas, licenciatura en Diseño de Comunicación/Diseño Gráfico o carreras a fines.

Áreas de experiencia:

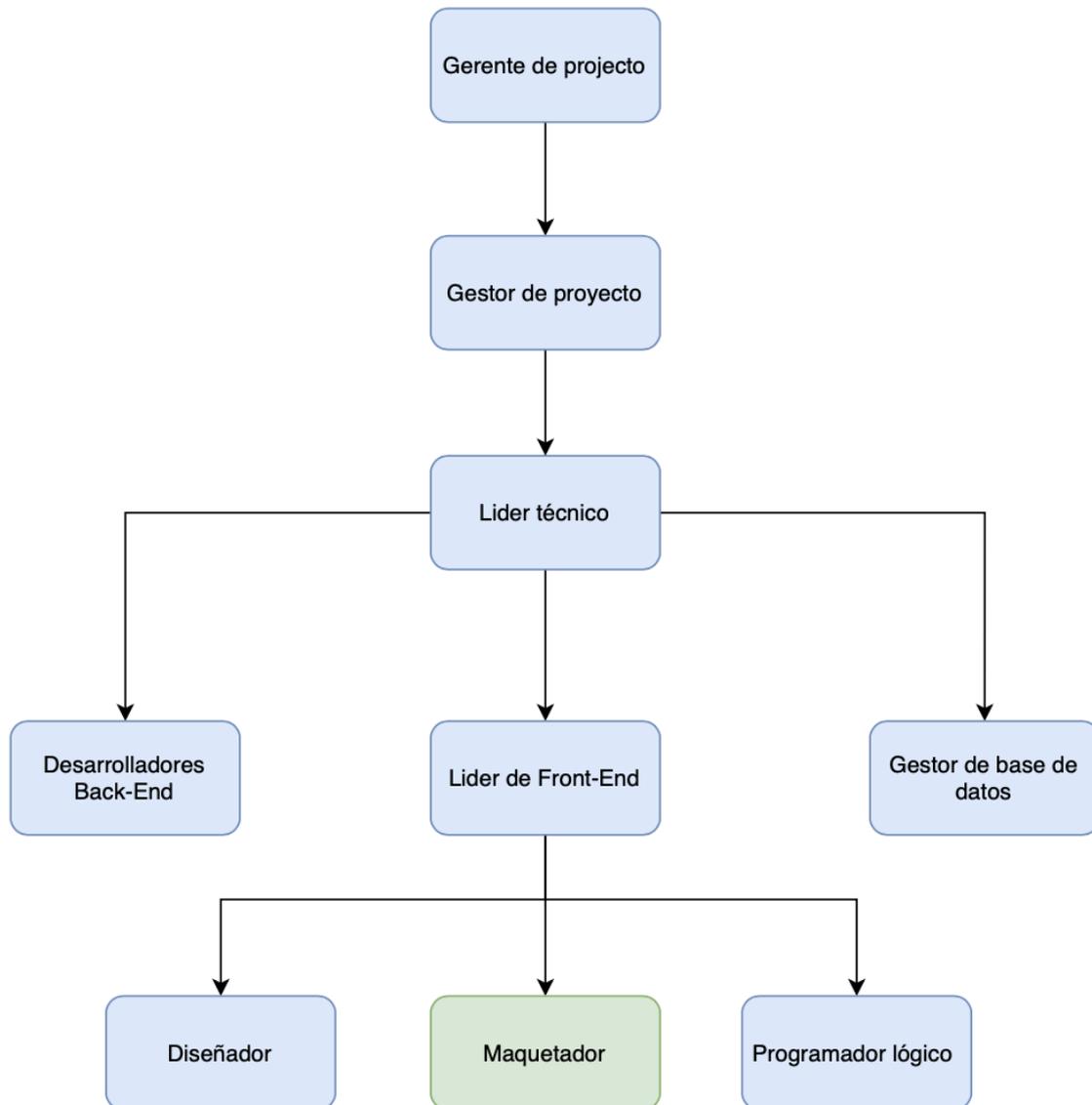
- UI/UX
- HTML
- CSS
- JavaScript
- TypeScript
- Angular
- Figma
- Abode XD
- Metodologías ágiles
- Azure DevOps

Tabla 15. Responsabilidades del maquetador

Responsabilidades	Periodicidad			
	Diaria	Semanal	Mensual	Ocasional
Crear biblioteca de componentes genéricos en Angular.	X			
Crear bibliotecas de clase de estilo.	X			
Replicar el prototipo de un sistema en HTML, CSS y JavaScript.	X			
Utilizar paleta de colores definido por el diseñador				X
Probar el software para garantizar su correcto funcionamiento				X
Trabajar con los equipos de desarrollo para asegurar que los diseños sean debidamente probados e implementados	X			
Trabajar eficazmente en varios proyectos al mismo tiempo	X			
Identificar y solucionar problemas de UI/UX (por ejemplo, capacidad de respuesta, diseño)	X			
Realizar ajustes a las aplicaciones web basados en los comentarios de los usuarios	X			
Realizar ajustes de diseño basados en los comentarios del líder técnico y gestores.	X			

Participar en capacitaciones y reuniones.			X	
Cumplir con las normas de estilo en cuanto a fuentes, colores, diseño e imágenes.	X			

Organigrama



Maquetador Web

DESCRIPCIÓN Y PERFIL DEL CARGO

2022

Descripción

Título del cargo	Área a la que pertenece
Maquetador Web	DPT
Cargo al que reporta	Sub-área a la que pertenece
Líder de Front-End	SPBC
Cargos que le reportan	Fecha de elaboración
N/A	20/06/2022

El maquetador web será el responsable de crear el diseño de las aplicaciones web en HTML, CSS y JavaScript, haciendo uso de Frameworks como Angular, Vue o React. Basándose en los diseños y prototipos entregados por el diseñador, asimismo deberá apegarse a los estándares, posiciones, tamaños y paleta de colores definidos previamente.

Requisitos administrativos

- Edad: en un rango de 21 a 35 años
- Sexo: Indistinto
- Domicilio: Tegucigalpa, FM

Condiciones sociales

Tabla 16. Condiciones sociales del programador lógico

Condiciones	Frecuentemente	Ocasional
Atiende Proveedores		X
Atiende Clientes	X	
Atiende Empleados	X	
Comparte sitio de trabajo		X
Requiere viajar		X
Jornada de trabajo		

Competencias Organizacionales**Tabla 17. Competencias Organizacionales del programador lógico**

Competencia	Grado
Orientación a la calidad	B
Orientación al servicio	A
Comunicación	A

Competencias Función/Técnicas**Tabla 18. Competencias Función/Técnicas del programador lógico**

Competencia	Grado
Comunicación eficaz	B
Ética	A
Auto aprendizaje	A
Innovación	A

Grados de Competencia:

A: Avanzado **B:** Experto **C:** Intermedio **D:** Básico

Conocimientos

Estudios:

Ingeniería en sistemas, licenciatura en Diseño de Comunicación/Diseño Gráfico o carreras a fines.

Áreas de experiencia:

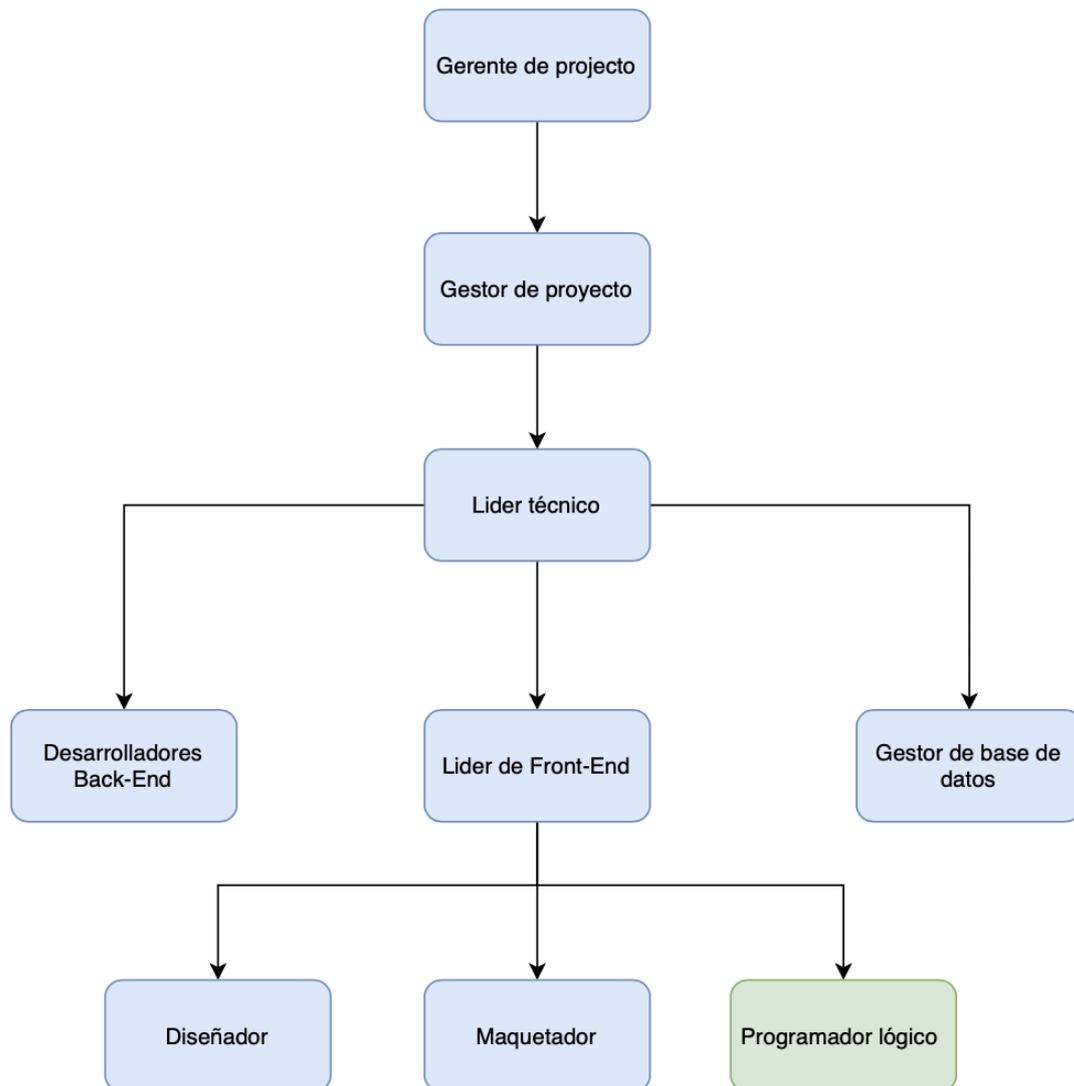
- HTML
- CSS
- JavaScript
- TypeScript
- Angular
- Vue
- React
- API REST
- Estructura de datos
- Ciberseguridad
- TFW
- Docker
- SignalR
- Figma
- Metodologías ágiles
- Azure DevOps

Tabla 19. Responsabilidades del programador lógico

Responsabilidades	Periodicidad			
	Diaria	Semanal	Mensual	Ocasional
Participar en las reuniones de levantamiento de requerimientos.	X			
Programar la lógica de las pantallas.	X			

Realizar imágenes Docker de la aplicación.	X			
Indagar sobre nuevas tecnologías.				X
Actualizar el Framework que se está utilizando.				X
Implementar seguridad a nivel de Front-End.	X			
Automatizar procesos para la implementación continua.	X			
Confirmar entradas y salidas según requerimientos del sistema.	X			
Realizar ajustes a las entradas y salidas de aplicaciones web basados en los comentarios de los usuarios	X			
Realizar ajustes a las entradas y salidas basado en los comentarios del líder técnico y gestores.	X			
Participar en capacitaciones y reuniones.			X	
Probar el software para garantizar su correcto funcionamiento	X			
Trabajar eficazmente en varios proyectos al mismo tiempo	x			
Trabajar con los equipos de desarrollo para asegurar que los diseños sean debidamente probados e implementados	X			

Organigrama



Una vez asignado el personal a prueba piloto y luego de haber definido el rol que realizará cada uno de los integrantes se capacitará al personal en áreas específicas mediante plataformas como Platzi, Coursera y documentación gratuita en línea, Ya que alguno de estos cuentan con un programa ilimitado ya pagado por la institución y también tienen cursos gratuitos esto no provocará el incremento del presupuesto para esta sección, asimismo se podrá capacitar al personal en áreas específicas a su rol detallando los de la siguiente manera:

- Diseñador:
 - UI/UX
 - HTML
 - CSS
 - JavaScript
 - Photoshop
 - Ilustrador
 - Figma
 - Abode XD
 - Metodologías ágiles
 - Azure DevOps

- Maquetador:
 - UI/UX
 - HTML
 - CSS
 - JavaScript
 - TypeScript
 - Angular
 - Figma

- Abode XD
- Metodologías ágiles
- Azure DevOps
- Programador lógico:
 - HTML
 - CSS
 - JavaScript
 - TypeScript
 - Angular
 - Vue
 - React
 - API REST
 - Estructura de datos
 - Ciberseguridad
 - TFW
 - Docker
 - SignalR
 - Figma
 - Metodologías ágiles
 - Azure DevOps

6.5 Medidas de control

Los indicadores que se utilizarán al realizar esta implementación serán en base al tiempo, rendimiento y satisfacción del usuario Negocio, ya que una de las principales ventajas de

fomentar la participación de Front-End es poder aumentar la satisfacción del usuario de Negocio desde el inicio del ciclo de vida del desarrollo de software provocando que el usuario realice la menor cantidad de solicitudes de cambio, para ello comenzaremos describiendo cuáles serán los indicadores que se utilizarán en la prueba piloto.

- Solicitudes de cambio: se evaluará la cantidad de solicitud de cambio al implementar la participación de estos roles, entre menor sea el número de solicitudes de cambio mayor será la satisfacción del cliente.
- Informe de pruebas técnicas: luego de que se termine el desarrollo de un producto entregable ya sea una pantalla, o una funcionalidad éste pasa a la etapa de pruebas técnicas para que el usuario valide el correcto funcionamiento y apariencia de este producto al finalizar el usuario deja un comentario de qué pruebas fueron aprobadas y cuáles fueron rechazadas o cual es necesitan un ajuste.
- Rendimiento: El rendimiento es un área fundamental para considerar un sistema como eficiente y al tener personas especializadas que ayuden a que las interfaces carguen de manera más rápida, poder validar entradas y salidas del sistema se podrá medir el tiempo de respuesta al cargar una interfaz y el tiempo de respuesta al validar ciertos parámetros de entrada.
- Eficiencia en los tiempos: Se realizarán métricas para contabilizar los tiempos en los cuales se crea un diseño, un prototipo, la maquetación de una pantalla, el flujo de un prototipo y la aplicación de la lógica de datos.

6.6 Proceso de implementación

Actividades a realizar:

- Identificación de habilidades y preferencias del equipo de desarrollo de Front-End de planta.
- Asignación de roles al personal del equipo de Front-End según sus habilidades y preferencias previamente identificadas.
- Capacitación del personal en áreas específicas.
- Identificación del módulo a desarrollar en prueba piloto.
- Solicitud y aprobación para prueba piloto en el módulo seleccionado a desarrollar.
- Reunión con el equipo de Negocio y personal técnico para la socialización de la nueva estructura de trabajo.
- Ejecución del ciclo de vida del desarrollo de software en el módulo seleccionado con la participación de los roles del equipo de Front-End.
- Evaluación semanal de indicadores.
- Discusión si se requiere contratación de nuevo personal.

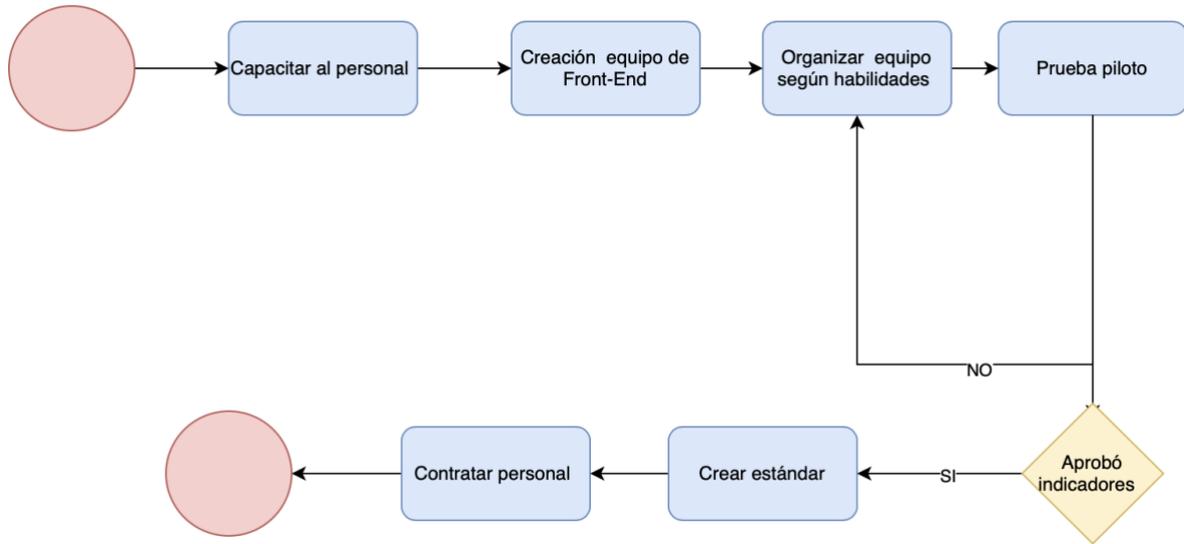


Figura 24: Proceso de implementación de roles en el Front-End
Fuente: Propia.

Bibliografía

1. Practice, G. A. B. (2019). *Itil Foundation, Itil 4 Edition: Spanish Translation* (4.^a ed.). Tso, the Stationery Office.
2. E. (2021, 8 marzo). *Tercera Revolución Industrial*. Economipedia. <https://economipedia.com/definiciones/tercera-revolucion-industrial.html>
3. *COBIT 2019 Framework: Introduction and Methodology*. Isaca.
4. *Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2014). Metodología de la investigación. McGraw Hill Interamericana.*
5. *TIOBE - The Software Quality Company. (2022). TIOBE Software BV.* <https://www.tiobe.com/tiobe-index/>
6. *Reglamento Sobre Gobierno Electrónico, PCM-086-2020, 26 DE SEPTIEMBRE DEL 2020*
7. *Código Penal de Honduras, DECRETO No. 93-2021, 1 DE NOVIEMBRE DEL 2021*
8. *digital, E. p. (s.f.). Obtenido de <https://www.docunecta.com/blog/que-es-la-gestion-de-proyectos>*
9. *Schwab, K. (2016). La cuarta revolución industrial.* <https://www.overdrive.com/search?q=111C4347-2121-4AE1-9B18-7C584195F118>
10. *Koho Software se une al programa global de socios de Agiloft: La asociación amplía el alcance*
11. Cabral, A. (2020, 5 marzo). *¿Cuáles son los Roles del desarrollo Front-end?* Azul School. <https://www.azulschool.net/cuales-son-los-roles-del-desarrollo-front-end/>

12. *ITIL / ITSM Service Desk Suite de Agiloft - ProQuest. (n.d.). Retrieved May 23, 2021, de*
<https://www.proquest.com/docview/1975576641/A1D0187A5CBB4E4APQ/1?accountid=35325>
13. *Front-end Developer Handbook 2019 - Learn the entire JavaScript, CSS and HTML development practice!* (2019). Front-End Master.
<https://frontendmasters.com/guides/front-end-handbook/2019/#3.2>
14. Rappin, N. (2021). *Modern Front-End Development for Rails: Hotwire, Stimulus, Turbo, and React*. Pragmatic Bookshelf.
15. Andrew Edkins, Joana Geraldi, Peter Morris y Alan Smith (2013) *Explorando el front-end de la gestión de proyectos, Engineering Project Organization Journal*
16. Jackson, C. (2019, 19 junio). *Micro Frontends*. martinowler.com. Recuperado 3 de marzo de 2022, de <https://martinowler.com/articles/micro-frontends.html>
17. *Cómo gestionar recursos de forma eficiente en tu empresa. (2019). work meter.*
Obtenido de <https://www.workmeter.com/blog/como-gestionar-recursos-de-forma-eficiente-en-tu-empresa/>
18. *Dedoimedo - Pencil. (2015). Dedoimedio.* <https://www.dedoimedo.com>

19. Amílcar Arantes, Luis Miguel D. F. Ferreira. (2021) [A methodology for the development of delay mitigation measures in construction projects](#). *Production Planning & Control*
20. Saiz Manzanares, M. C. E. L. (2013). *Observacion sistematica e investigacion en contextos educativos*. Universidad De Burgos. <http://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3217332>
21. Edix, R. (2022, 5 abril). *Front End Developer: ¿cuáles son sus funciones y salidas?* Edix España. <https://www.edix.com/es/instituto/front-end-developer/>
22. Terry Williams, Hang Vo, Knut Samset & Andrew Edkins (2019) The front-end of projects: a systematic literature review and structuring, *Production Planning & Control*, 30:14, 1137-1169, DOI: [10.1080/09537287.2019.1594429](https://doi.org/10.1080/09537287.2019.1594429)
23. Andrew Edkins, Joana Geraldi, Peter Morris y Alan Smith (2013) *Explorando el front-end de la gestión de proyectos*, *Engineering Project Organization Journal*

Anexo 1

Guía para entrevista

¿Qué puesto desempeña en la institución?

¿Cuántos años de experiencia tiene en el puesto?

¿Cuál es el área en la que usted se especializa?

¿Ha participado en el desarrollo de un sistema, siguiendo el ciclo de vida del desarrollo desde su etapa inicial hasta la final?

¿En que etapas se ha involucrado y como ha sido su experiencia?

¿Que etapas cree que son las que con llevan mayor trabajo?

¿Que etapas cree que son las que tiene una mayor prioridad?

¿Que etapas considera que pueden mejorarse?

¿Qué propone para mejorar estas etapas?

¿En base a su experiencia que es Front-End?

¿Cuántas personas laboran en el área de Front-End?

¿Qué actividades realizan los integrantes del componente de Front-End?

¿Considera usted que el equipo de Front-End podría realizar otras actividades y cuales serian?

¿Que beneficios ocasionara que el equipo de Front-End realice estas actividades?

Tomando en consideración las funciones propuestas, en base a su experiencia ¿consideraría que la institución cuenta con la capacidad en cuanto a colaboradores para poder segmentar estas funcionalidades y puedan centrarse en el funcionamiento de su rol específico?

¿En qué etapas de SDLC involucran al personal de Front-End?

¿Qué tecnología utiliza el componente de Front-End?

¿Quién coordina al equipo de Front-End?

¿Quien establece las actividades de el equipo de Front-End?

- ¿Existe algún estándar que deban seguir los desarrolladores de Front-End?
- ¿Qué herramientas utilizan los integrantes del equipo de Front-End?
- ¿Qué herramientas o Framework que no son de paga tienen permitido utilizar?

Anexo 2

Guía consulta a experto

- ¿Qué puesto desempeña en la institución?
- ¿Cuántos años de experiencia tiene en el puesto?
- ¿Cuál es el área en la que usted se especializa?
- ¿En base a su experiencia como ha funcionado implementar la nueva estructura al dividir el componente de Back-End y Front-End?
- ¿Cuántas personas integran cada uno de estos equipos?
- ¿Ha obtenido una mejoría en el rendimiento de las aplicaciones?
- ¿Usted involucra a el líder de cada uno de estos equipos en las reuniones con los usuarios?
- ¿Que resultados obtenidos al incluir al líder del grupo de Front-End en el levantamiento de requerimientos con los usuarios?
- ¿Que opiniones de los usuarios ha tenido al mostrar mejor calidad en los diseños??
- ¿Que recomienda al momento de implementar un equipo de desarrollo de Front-End

Anexo 3

Entrevistas a personal del BCH, banca privada e instituciones del gobierno de Honduras que laboran en el área de TI.

Entrevista 1

¿Qué puesto desempeña en la institución?

Desarrollador de Front-End

¿Cuántos años de experiencia tiene en el puesto?

4 años

¿Cuál es el área en la que usted se especializa?

Me especializo en la experiencia y diseño de usuario

¿Ha participado en el desarrollo de un sistema, siguiendo el ciclo de vida del desarrollo desde su etapa inicial hasta la final?

Si

¿En que etapas a se ha involucrado y como ha sido su experiencia?

Me he visto involucrado en el desarrollo y en las pruebas de un sistema, es un poco tedioso como desarrollador Front-End que los usuarios están desarrollando múltiples solicitudes de cambio debido a que los diseños que tienen los requerimientos del sistema no son nada específicos, detallados, o nada parecidos a lo que los usuarios esperan del sistema.

¿Que etapas cree que son las que con llevan mayor trabajo?

Las áreas que con lleva un mayor trabajo son la del análisis del sistema diseño y desarrollo.

¿Que etapas cree que son las que tiene una mayor prioridad?

Las etapas que tienen una mayor prioridad en mi opinión debería ser el estudio de viabilidad, el análisis del sistema y el desarrollo, no es que las demás etapas no tengan importancia o no se les de una prioridad, pero estas tres etapas previamente mencionadas son las que pueden definir el éxito o el fracaso de un sistema.

¿Que etapas considera que pueden mejorarse?

La etapa que yo considero que podría mejorarse es la del análisis del sistema que aunque con lleve mayor cantidad de tiempo tanto del equipo de tecnología como de los usuarios de Negocio se crearon requerimientos con un nivel detallado mucho mayor que incluyan diagramas de proceso ejemplo de la salidas y de las entradas que tendrá el sistema, diseños realmente detallados como se verá el sistema al desarrollarse y qué salidas tendrá asimismo que entradas deberá ofrecer este sistema y esto sólo como base se pueden ir detallando otros aspectos que incluyen los requerimientos del sistema esto provocará que podamos tener una mayor planificación a un detalle más específico poder aplicar la metodología de Scrum y dar entregables en menor tiempo y más certeros.

¿Qué propone para mejorar estas etapas?

Yo propongo que se siga el estándar interno para el desarrollo de sistemas, pero utilizando tecnologías actuales es decir en ese estándar una sección no solicita que creamos diseños de las pantallas pero que esto sea con tecnologías modernas como Adobe XD, Otro

ejemplo podría ser diagramar los procesos en herramientas como bisagras y que tengamos una comunicación más fluida con el equipo de tecnología y los usuarios de negocio.

¿En base a su experiencia que es Front-End?

Es un equipo que se encarga de desarrollar, diseñar y probar la apariencia de un sistema, asimismo la experiencia que el usuario tendrá desde solicitar el sistema hasta tenerlo en un ambiente de producción.

¿Cuántas personas laboran en el área de Front-End?

4 personas.

¿Qué actividades realizan los integrantes del componente de Front-End?

Actualmente son actividades muy diversas ya que estamos comenzando a perfeccionar la arquitectura de micro servicios de parte de Back-End en un repositorio monolítico para Front-End entre las principales actividades está el diseño de pantallas, la maquetación de estos diseños, el prototipo piado, el desarrollo en angular y las pruebas de estas pantallas.

¿Considera usted que el equipo de Front-End podría realizar otras actividades y cuales serian?

Sí podrían participar en otras actividades como en las reuniones con el usuario de negocio y que pueda tener una opinión referente a que es posible o no en el desarrollo de un sistema.

¿Que beneficios ocasionara que el equipo de Front-End realice estas actividades?

Ocasionaría que los ajustes a las funcionalidades o al sistema completo sean menores y poder entregar componentes de mejor calidad.

Tomando en consideración las funciones propuestas, en base a su experiencia

¿consideraría que la institución cuenta con la capacidad en cuanto a colaboradores para

poder

segmentar estos y puedan centrarse un el funcionamiento de su rol específico?

Si es posible.

¿En qué etapas de SDLC involucran al personal de Front-End?

Involucrar al personal de FrontEnd serían en análisis del sistema la planificación, el diseño, el desarrollo, las pruebas y la implementación.

¿Qué tecnología utiliza el componente de Front-End?

Firma, angular, visual Code, Visual Studio, Azure, Docker, entre otras.

¿Quién coordina al equipo de Front-End?

El líder técnico.

¿Quien establece las actividades de el equipo de Front-End?

El gerente del proyecto.

¿Existe algún estándar que deban seguir los desarrolladores de Front-End?

Aun no, pero se esta trabajando en un estándar que será enfocado al diseño y experiencia.

ANEXO 4

Guía consulta a experto

¿Qué puesto desempeña en la institución?

Gestor de proyectos

¿Cuántos años de experiencia tiene en el puesto?

15

¿Cuál es el área en la que usted se especializa?

Me he especializado en la gestión de proyectos de la cual tengo una maestría en administración de proyecto y una certificación del PMP.

¿En base a su experiencia como ha funcionado implementar la nueva estructura al dividir el componente de Back-End y Front-End?

Ha funcionado de manera exitosa desde distintos aspectos, ya que la responsabilidad se segmento en distintas áreas y es mucho más eficiente tener expertos en un área, que tener un experto en todas las áreas. Como resultado hemos tenido reducción en los tiempos productos de mayor calidad tecnologías innovadoras ya que se enfocan e investigan en su componente, ya sea Back-End, Front-End arquitectura o infraestructura de una aplicación.

¿Cuántas personas integran cada uno de estos equipos?

Back 6 personas, Front-End 4 personas, arquitectura 2 personas e infraestructura 2 personas.

¿Ha obtenido una mejoría en el rendimiento de las aplicaciones?

Sí, los micro servicios nos han ayudado en la respuesta más rápida de la información extraer de base de datos hasta poder mostrárselas al usuario e incluso estamos analizando ahora los micro servicios para Front-End, ya que para Back-End ya están implementados y funcionando.

¿Usted involucra a el líder de cada uno de estos equipos en las reuniones con los usuarios?

Si

¿Que resultados obtenidos al incluir al líder del grupo de Front-End en el levantamiento de requerimientos con los usuarios?

Cuándo se forman equipos de trabajo y no solamente grupos de trabajo, se consigue que todos ayuden a impulsar un proyecto hacia adelante y cuando nuestro líder del equipo de Front-End participa en estas reuniones nos da una opinión mucho más eficiente de cómo poder construir la apariencia de un sistema como poder mostrar la información de una manera más óptima y nos ayuda diseñando en tiempo real como podría ver si el sistema al finalizar pero no sólo como una imagen sino que en una semana luego de la reunión nos muestra un prototipo de qué flujo pueda tener este sistema lo que provoca que los sistemas sean más novedosos que los usuarios puedan opinar si es lo que ellos esperan o no y esto nos ayuda a incrementar la calidad de nuestros sistemas.

¿Que opiniones de los usuarios ha tenido al mostrar mejor calidad en los diseños?

Al inicio los usuarios se cohibían porque estábamos acostumbrados a tener reuniones

entre jefes una jerarquía alta y al ver que personal nuevo comenzaba participar en esas reuniones fue difícil, pero al ver que teníamos una mejor calidad que ofrecer sus opiniones eran muy satisfactorias ya que miraban que eran cambios para mejorar y actualmente todos nuestros usuarios están muy satisfechos de la manera que estamos trabajando, comunicación, calidad y otros aspectos.

¿Que recomienda al momento de implementar un equipo de desarrollo de Front-End?

Recomiendo que busquen a un experto en el área que le guste este componente y que ayude a indagar e implementar tecnologías actuales que puedan ir mejorando todos los componentes y que puedan formar un equipo de trabajo que ayude a mejorar todas las áreas y etapas en el ciclo de vida del desarrollo de un sistema y de esta manera todos irán empujando el proyecto hasta ofrecer la mayor calidad en el menor tiempo posible.