



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ingeniería Escuela de Ciencias y Sistemas

**ANALISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE  
PARA EMPRESAS EN EL SECTOR GUATEMALTECO**

**Christopher Jhoanis Soto Cifuentes**

Asesorado por el Ingeniero Jhonatan Wilfredo Pú Morales

Guatemala, marzo de 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ANALISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE  
PARA EMPRESAS EN EL SECTOR GUATEMALTECO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR

**CHRISTOPHER JHOANIS SOTO CIFUENTES**

ASESORADO POR INGENIERO JHONATAN WILFREDO PÚ MORALES

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, MARZO DE 2023

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martinez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz Gonzalez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANA	inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Álvaro Giovanni Longo Morales
EXAMINADOR	Ing. Alvaro Obrayan Hernández García
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **ANALISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE PARA EMPRESAS EN EL SECTOR GUATEMALTECO**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 23 de agosto 2021.



**Christopher Jhoanis Soto Cifuentes**

Guatemala, 3 de diciembre de 2022

Ingeniero  
**Carlos Alfredo Azurdia**  
Coordinador de Privados y Trabajos de Tesis  
Escuela de Ingeniería en Ciencias y Sistemas  
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **CHRISTOPHER JHOANIS SOTO CIFUENTES** con carné **201602569** y CUI **2988 16547 0101** titulado **"ANÁLISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE PARA EMPRESAS EN EL SECTOR GUATEMALTECO"**, lo he revisado y luego de corroborar que el mismo se encuentra finalizado y que cumple con los objetivos propuestos en el respectivo protocolo, procedo a la aprobación respectiva.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
*Jhonatan Wilfredo Pú Morales*  
Ingeniero en Ciencias y Sistemas  
Colegiado 15,628  
**Ing. JHONATAN WILFREDO PÚ MORALES**  
Colegiado No. 15628



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 26 de enero de 2023

Ingeniero  
**Carlos Gustavo Alonzo**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **CHRISTOPHER JHOANIS SOTO CIFUENTES** con carné **201602569** y CUI **2988 16547 0101** titulado **“ANÁLISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE PARA EMPRESAS EN EL SECTOR GUATEMALTECO”**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA

LNG.DIRECTOR.061.EICCSS.2023

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del Asesor, el visto bueno del Coordinador de área y la aprobación del área de lingüística del trabajo de graduación titulado: **ANALISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE PARA EMPRESAS EN EL SECTOR GUATEMALTECO**, presentado por: **Christopher Jhoanis Soto Cifuentes**, procedo con el Aval del mismo, ya que cumple con los requisitos normados por la Facultad de Ingeniería.

“ID Y ENSEÑAD A TODOS”

Ing. Carlos Gustavo Alonzo  
Director

Escuela de Ingeniería en Ciencias y Sistemas

Msc. Ing. Carlos Gustavo Alonzo  
Director

Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, marzo de 2023





Decanato  
Facultad de Ingeniería  
24189101- 24189102  
secretariadecanato@ingenieria.usac.edu.gt

LNG.DECANATO.OI.288.2023

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **ANÁLISIS DE LA VIABILIDAD EN LA IMPLEMENTACIÓN DE APLICACIONES LOW-CODE PARA EMPRESAS EN EL SECTOR GUATEMALTECO**, presentado por: **Christopher Jehanis Soto Cifuentes**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.



IMPRÍMASE:

Inga. Aurelia Anabela Cordova Estrada  
Decana



Guatemala, marzo de 2023

AACE/gaoc

## **ACTO QUE DEDICO A:**

### **Dios y a la vida**

Que me dieron la oportunidad de tener una familia que me apoyó en todo momento.

### **Mi mamá**

Que es el amor de mi vida y que siempre estuvo pendiente de mí, aguantando mis frustraciones y mi mal humor, que me apoyo en cada desvelo y en cada decisión, que siempre confió en mí, y que sin ella no seria la persona que soy.

### **Mi papá**

Que trabajó siempre y llevo el sustento a la casa para que mi mamá pudiera criarme de la mejor manera.

### **Mi abuela**

Que siempre estuvo presente y es como mi segunda madre.

### **Mis amigos**

Con los que compartí aulas, cursos, frustraciones y felicidad a lo largo de la carrera.

## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de Guatemala**

Que me permitió tener la oportunidad de recibir educación superior.

**Mis amigos**

Que estuvieron presentes, que me dieron animos y apoyo para lograr llegar a este punto.

## ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES .....	V
LISTA DE SÍMBOLOS .....	VII
GLOSARIO .....	IX
RESUMEN .....	XI
OBJETIVOS.....	XIII
INTRODUCCIÓN.....	XV
1. LOW-CODE Y TRANSFORMACION DIGITAL .....	1
1.1. ¿Qué es low-code? .....	1
1.2. Características del low-code.....	2
1.2.1. Código reducido.....	2
1.2.2. Entorno amigable.....	3
1.2.3. Innovación escalable .....	3
1.2.4. Mayor alcance.....	3
1.3. Ventajas de las plataformas low-code .....	4
1.3.1. Disminución de tiempos de entrega .....	4
1.3.2. Menor inversión inicial .....	4
1.3.3. Tiempos menores en cambios .....	4
1.3.4. Flujos fáciles de entender.....	5
1.3.5. Herramientas visuales .....	5
1.4. Beneficios de utilizar las plataformas low-code.....	5
1.5. ¿Por qué low-code?.....	7
1.6. Transformación digital.....	7
1.6.1. Importancia de la transformación digital.....	10

2.	FACTORES QUE INTERRUNPEN EN LAS ENTREGAS DE LOS PROYECTOS DE IT .....	11
2.1.	Requerimientos deficientes y ambiguos.....	11
2.2.	Calculos ineficientes en tiempos de desarrollo .....	13
2.2.1.	Implementacion de Planning Poker.....	14
2.2.1.1.	Escala de estimación .....	14
2.2.1.2.	Reglas del Planning Poker.....	14
2.3.	Falta de implementación de metodologías ágiles de desarrollo o implementaciones incorrectas.....	16
2.4.	Metodologías ágiles .....	17
2.4.1.	Principios de metodologías ágiles.....	17
2.5.	Principales metodologías ágiles .....	20
2.5.1.	Scrum.....	21
2.5.2.	XP .....	23
2.5.3.	Crystal Clear .....	24
2.5.4.	Kanban.....	26
2.6.	Requisitos ágiles.....	28
2.6.1.	Principales técnicas para toma de requisitos en las metodologías ágiles .....	29
2.6.2.	Historias de usuario .....	32
2.7.	Metodología Rup (Rational Unifield Process).....	33
2.7.1.	Caracteristicas de Rup .....	33
2.7.2.	Elementos metodología Rup .....	34
3.	HERRAMIENTAS PARA IMPLEMENTAR LOW-CODE .....	37
3.1.	Uipath.....	37
3.1.1.	Que es uipath .....	37
3.1.2.	Beneficios de implementar RPA.....	39
3.1.3.	Ventajas de Uipath .....	40

3.2.	Creatio.....	42
3.2.1.	Que es Creatio .....	42
3.2.1.1.	Studio Creatio .....	42
3.2.1.2.	Sales Creatio.....	42
3.2.1.3.	<i>Marketing</i> Creatio.....	43
3.2.1.4.	Service Creatio.....	43
3.2.2.	Beneficios de implementar Creatio .....	45
3.3.	Ventajas de Creatio.....	47
3.4.	OutSystems.....	48
3.4.1.	Que es OutSystems .....	48
3.4.2.	Entornos de OutSystems .....	49
3.4.3.	Ventajas de Outsystems .....	51
4.	IMPLEMENTACION CON CREATIO .....	55
4.1.	Creación de la cuenta .....	55
4.2.	Creación del nuevo proceso.....	56
4.2.1.	Notacion Creatio.....	58
5.	ANALISIS DE RESULTADOS.....	61
5.1.	Resultados obtenidos .....	61
5.1.1.	Empresa 1 .....	61
5.1.2.	Empresa 2 .....	62
	CONCLUSIONES .....	65
	RECOMENDACIONES.....	67
	REFERENCIAS .....	69



## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1.	Guadrante de Gartner 2019 sobre aplicaciones low-code .....	2
2.	Ayuda del low-code en la demanda de desarrolladores .....	6
3.	Pasos para evitar el retroceso digital.....	8
4.	Cartas de Planning Poker .....	16
5.	Herramienta de planificación empresarial agil Gartner 2018 .....	19
6.	Flujo de SCRUM .....	23
7.	Flujo de XP.....	24
8.	Propiedades Crystal Clear .....	25
9.	Tablero Kanban.....	27
10.	Fases del método RUP.....	36
11.	RPA en el cuadrante de Gartner .....	41
12.	Creatio en el cuadrante de Gartner .....	44
13.	Connectos de Creatio .....	46
14.	Dashboards de Creatio .....	47
15.	Características OutSystems.....	50
16.	OutSystems en el cuadrante de Gartner .....	52
17.	Login Creatio.....	55
18.	Página inicial Nuevo Proceso Creatio .....	56
19.	Nombrar al nuevo proceso Creatio.....	57
20.	Proceso vacío en Creatio.....	57
21.	Símbolos de evento .....	59
22.	Símbolos de tarea.....	60
23.	Proceso terminado Creatio .....	60

24.	Costo de implementación empresa 1 .....	61
25.	Tiempos sin Low-Code empresa 2 .....	63
26.	Tiempos con Low-Code empresa 2 .....	63

## **TABLAS**

I.	Escapa Planning Poker .....	14
II.	Historia de usuario .....	33
III.	Compatibilidad OutSystems .....	49

## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>AUP</b>	Agile Unified Process
<b>BD</b>	Base de datos
<b>CRUD</b>	Create, Read, Update, Delete
<b>XP</b>	Extreme Programming
<b>LC</b>	Low-Code
<b>Mb</b>	Mega Bite
<b>RPA</b>	Robotic Process Automation



## GLOSARIO

<b>Cuadrante de Gartner</b>	Listado que representa las tendencias de alguna tecnología en específico.
<b>Desarrollo</b>	Funcionalidad específica de un sistema.
<b>Low-code</b>	Herramientas que ayudan a realizar procesos similares a los que realizan los desarrolladores de software con menor cantidad de dificultad y de código.
<b>Metodología</b>	Es el conjunto de actividades teóricas y sistemáticas que se utilizan para estructurar, planificar y controlar la elaboración de alguna tarea.
<b>Requerimiento</b>	Se define como la funcionalidad que un cliente espera de un software terminado.
<b><i>Sprint</i></b>	Periodo de tiempo dentro del cual debe cumplirse una cantidad de tareas anteriormente pactadas.



## RESUMEN

Debido a la dificultad de entregar un proyecto a tiempo, que es una realidad que muchos viven por diferentes razones, ya sea por una mala administración del control de tiempos y entregas, o en ocasiones los desarrolladores no entienden bien el giro del negocio, de la empresa, o los interesados (*sponsors*), no saben expresar lo que realmente quieren. Por esa razón los proyectos tienden a atrasarse y a no estar listos en el tiempo que se tenían planeados. Por ejemplo un desarrollador aprende sobre contabilidad o medicina para poder desarrollar el sistema que solicitan, otra situación que se da mucho para que un proyecto no esté terminado a tiempo, es que los requerimientos no son tomados de la mejor manera, esto lleva a las aplicaciones low-code en las que personas no necesariamente del área de Sistemas o Desarrollo, pueden incluirse para poder aportar a dicha aplicación, como lo pueden ser paginas web desarrolladas con drag and drop donde las personas solo eligen lo que quieren y lo colocan.

Otra razón que se observa es que la demanda de desarrollos es mucho más grande que la de desarrolladores, y contar con más desarrolladores podría ayudar a que se cumpla con más trabajos y se terminen a tiempo.



## **OBJETIVOS**

### **General**

Evaluar la viabilidad de la implementación de las aplicaciones low-code en empresas guatemaltecas, de tal forma que los proyectos se adapten a las necesidades que las empresas requieran.

### **Específicos**

1. Presentar las ventajas de implementar low-code en las empresas guatemaltecas.
2. Ayudar a las empresas a poder diseñar una propuesta de low-code que satisfaga sus necesidades en distintos giros de negocio.
3. Comparar la productividad y desempeño de la empresa antes y después de utilizar una herramienta de low-code.



## INTRODUCCIÓN

En el presente trabajo de graduación se encuentra una serie de análisis y situaciones donde se define que, implementar tecnologías low-code para la inclusión de *bots* y procesos en las empresas del sector guatemalteco, representa una mejora, debido a que agiliza procesos, disminuye el error humano, y ayudará a que en las entregas de proyectos de tecnología no se atrasen los entregables, o existan malos entendidos entre requerimientos.

Además, se demuestra como estas implementaciones ayudan a que las empresas ahorren dinero y logren aumentar su productividad sin tener que recurrir a inversiones altas, y que al utilizar estas herramientas muchas personas lograrían desarrollar lo que necesiten sin necesidad de acudir al departamento de Desarrollo o IT, que son quienes dan soporte, implementan, y mantienen la información de los sistemas. Administran la seguridad de las redes, computadoras y todo lo que tiene que ver con informática.



# **1. LOW-CODE Y TRANSFORMACION DIGITAL**

En este capítulo se expone el concepto general de que es low-code y su importancia en la transformación digital para comprender de una mejor forma las utilidades y ventajas que ofrecen estar actualizados hoy en día. Se da a conocer las ventajas de dicha tecnología y el porque es una buena opción de implementación en las empresas del sector guatemalteco para disminuir tiempos de entrega, errores en tomas de requerimientos y en aumentar el entendimiento y desemepeño de todos los interesados en la parte de los desarrollos a realizar.

## **1.1. ¿Qué es low-code?**

Se define al low-code como una herramienta que permite crear, modificar, mantener y desarrollar software de una manera eficiente y sencilla, esto con el fin de que personas ajenas al desarrollo de software puedan contribuir y generar proyectos de software para poder cumplir la demanda de software a nivel general, esto agiliza el entendimiento del giro de negocio, porque los desarrolladores no tardarán tanto en entender el requerimiento, si no que las mismas personas que quieren el requerimiento podrán aportar.

Figura 1. **Guadrante de Gartner 2019 sobre aplicaciones low-code**



Fuente: Creatio. (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

## 1.2. Características del low-code

Para entender como se comportan las aplicaciones de low-code se describen algunas de sus características más relevantes.

### 1.2.1. Código reducido

Desarrollar con aplicaciones low-code permite que las personas puedan realizar pequeños desarrollos que posteriormente pueden ir evolucionando con el menor código posible escrito por ellos.

### **1.2.2. Entorno amigable**

Las plataformas para low-code proporcionan un entorno de desarrollo más gráfico que los IDEs normales, en donde se pueden desarrollar aplicaciones de software por medio de interfaces gráficas y configuraciones, teniendo en cuenta que las personas que las utilizan no son desarrolladores de software experimentados.

Muchas de las plataformas utilizan el famoso drag and drop (arrastrar y soltar), así como plantillas predefinidas, esto garantiza que empleados o empresas sin experiencia puedan colaborar en los proyectos de desarrollo, disminuyendo la presión sobre el equipo de desarrollo y de IT.

### **1.2.3. Innovación escalable**

Las aplicaciones low-code permiten que mientras se desarrollan se vayan desdibujando mejor, es decir una idea original puede ir madurando en el camino y cubrir necesidades que no fueron tomadas en cuenta al inicio, esto sin necesidad de alterar tanto los requerimientos iniciales del desarrollo.

### **1.2.4. Mayor alcance**

Debido a que dichas plataformas utilizan la interfaz gráfica y configuraciones para desarrollar y crear contenido no es necesario que se enfoque en un solo entorno como lo puede ser: móvil, web o de escritorio es decir las personas podrán implementarlas en las distintas vías o entornos que deseen.

### **1.3. Ventajas de las plataformas low-code**

Una vez que se entiende que son y cuales son las características de las plataformas low-code se tiene una idea de cual será utilidad y ventajas en su implementación en las empresas del sector guatemalteco, algunas de las ventajas más importantes de estas plataformas son:

#### **1.3.1. Disminución de tiempos de entrega**

Una de las ventajas principales de la implementación de las aplicaciones low-code es que no necesariamente el equipo de desarrollo o de IT tendrá que hacerlo, existiran proyectos en donde personas de áreas ajenas a la tecnología podrían crear sus propias paginas web, o programas para poder disminuir los tiempos de entrega de procesos dentro de sus entidades o empresas.

#### **1.3.2. Menor inversión inicial**

Al contar con una gran cantidad de herramientas visuales y de plantillas no es necesario invertir tanto en capacitaciones o certificaciones en algún lenguaje de programación o tecnología específica porque será mucho más sencillo utilizar estas tecnologías.

#### **1.3.3. Tiempos menores en cambios**

Al no ser aplicaciones desarrolladas desde cero permitira que estas aplicaciones puedan modificarse rapidamente con ayuda de los asistentes de configuración y los asistentes gráficos que estas aplicaciones traen incluidos, por lo que un cambio de algún *sponsor* o cliente puede ser realizado en menor tiempo utilizando la tecnología low-code.

#### **1.3.4. Flujos fáciles de entender**

Los flujos para utilizar las aplicaciones low-code son fáciles de entender y cualquier persona podrá arrastrar y soltar componentes que desee utilizar en su solución, y configurar el funcionamiento que le desee implementar.

#### **1.3.5. Herramientas visuales**

Las aplicaciones low-code en vez de funcionar con grandes sintaxis de código funcionan con herramientas visuales lo que garantiza que personas ajenas a las ciencias de la computación como lo son analistas, encargados de procesos y operaciones o bien jefes de área puedan utilizarlas y aportar.

#### **1.4. Beneficios de utilizar las plataformas low-code**

Uno de los beneficios principales de utilizar estas tecnologías es que los usuarios ajenos al desarrollo pueden ingresar sus requerimientos directamente al entorno de la plataforma low-code. Esto ayuda a que los desarrolladores tengan más tiempo para enfocarse en trabajos que dependan un desarrollo desde cero o bien mantenimientos a softwares que no se realicen con las plataformas de low-code.

Otros de los beneficios de estas tecnologías pueden ser:

- Las personas que los utilizan pueden tener menor habilidad en programación y computación y aun así realizar un desarrollo.

- Las empresas se volverán menos dependientes de los profesionales de la programación y esto ayudara a resolver los problemas con la demanda de desarrolladores.
- Mejora la colaboración, agilidad y productividad de los procesos al implementarse una ayuda de software en procesos de las empresas

Por otra parte, si se logra implementar inteligencia artificial a las plataformas desarrolladas con low-code estas tendrán un alto potencial para realizar de una mejor manera procesos comerciales con tareas accesibles y simples.

Figura 2. **Ayuda del low-code en la demanda de desarrolladores**



Fuente: Creatio, (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

### **1.5. ¿Por qué low-code?**

Una vez se entiende lo que es la tecnología de low-code y se comprende el problema de los escasos desarrolladores para todo el trabajo que se tiene en el área de la computación, es necesario hacer algo y que más que poder hacer un trabajo de desarrollo sin la necesidad de tener que ser desarrollador esta es la maravilla del low-code cualquier persona puede usarlo y crear cosas.

Un ejemplo bastante claro se vivió con el COVID-19 en el año 2020, las personas migraron sus servicios a internet y a hacerlo todo de una manera digital porque la movilización fue limitada, entonces las empresas que querían sobrevivir tenían que estar en internet, y muchas optaron por crear sus paginas web, sus aplicaciones para teléfonos móviles esto implicaba un gran trabajo para los especialistas de software, implementar pasarelas de pago, darle mantenimiento a paginas obsoletas, migrar información, darle soporte a bases de datos antiguas.

### **1.6. Transformación digital**

Se puede definir a la transformación digital como la aplicación de todas las capacidades digitales e informáticas a procesos, productos y activos de una empresa para mejorar su utilidad y eficiencia, mejorar el valor que se brinda a los clientes y para disminuir riesgos, también sirve para descubrir nuevas fuentes de ingresos, también se conoce como transformación digital a la integración de la tecnología en todas las áreas de una empresa dando un cambio drástico a la manera en la que opera.

Se dice que la transformación digital es la integración de tecnología digital puede implicar la reelaboración de protocolos, estrategias y productos

dentro de las organizaciones mediante el aprovechamiento de la tecnología digital.

En pocas palabras la transformación digital requiere una revisión y reestructuración de la mayoría de las áreas dentro de la operación de la organización, para mejorar flujos de trabajo, habilidades de sus integrantes y procesos de discusión en la dirección de dicha organización. La transformación digital es una herramienta que ayuda a una organización a seguir el ritmo de su competencia y a la demanda emergente de sus clientes, manteniéndolos actualizados y en el futuro, la transformación digital es necesaria para que cualquier empresa, organización o institución que busque sobrevivir en el futuro.

Figura 3. **Pasos para evitar el retroceso digital**



Fuente: Creatio, (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

Una vez que se entiende la importancia de la transformación digital se debe tomar en cuenta que al tratarse de temas tecnológicos estos avanzan con mucha rapidez y no hay que quedarse atrás o bien retroceder en estos procesos, por lo que algunos consejos que se deben seguir son:

- Haz una pausa, detenerse y tomar el tiempo para evaluar como se encuentra la organización y que necesita en ese momento es algo que se debe hacer si se quiere mantener a flote, una evaluación a la mitad de la transformación puede ayudar a que los interesados recopilen perspectivas de lo que se quiere lograr con la transformación, evaluar datos y metas que se desean cumplir, se recomienda que todos los interesados participen en estas evaluaciones y que se verifiquen los modelos del negocio.
- Realizar una evaluación, una vez que se identifica a donde se quiere llegar con la transformación digital (conservar, revertir o reinventar), se evalúa la necesidad de los sistemas y la tecnología que se desea implementar y cuales serán sus funciones.
- Enseña, una vez que se realizo la evaluación se logra identificar los cambios necesarios que la organización debe realizar para conservar, revertir o reinventar los compartimientos y modelos que se tenían en un inicio, se aconseja diseñar planes de acción claros para cada modificación que se desea llevar a cabo en la organización, así como asignar la responsabilidad con expectativas que garanticen que estos cambios se van a cumplir.

### **1.6.1. Importancia de la transformación digital**

La mayoría de organizaciones y empresas que perciben a la transformación digital como una diferencia de tecnología sobre lo que ya tienen, junto con un cambio en la ingeniería de su giro de negocio por lo general resultan alejándose del verdadero enfoque y poder que pueda tener de la transformación digital.

Hablar en terminos digitales hace referencia a convertir procesos en datos, esto permite que las organizaciones puedan ver como se esta comportando su operación a travez de datos reales que dan lugar a tener objetivos y resultados.

A medida que surgen los datos, la capacidad de estudiarlos, organizarlos y entenderlos, esos datos con indicadores pueden hacer saber como esta la posicion de la empresa. Una organización puede no ser solo más eficiente sino también cambiar lo que hace.

## **2. FACTORES QUE INTERRUNPEN EN LAS ENTREGAS DE LOS PROYECTOS DE IT**

En este capítulo se tratarán las principales razones por las que no se logran entregar a tiempo los proyectos de IT, o bien por los que estos sufren atrasos o complicaciones en su desarrollo, se habla de como mejorarlo y de como al implementar el low-code muchas de estas razones disminuyen.

### **2.1. Requerimientos deficientes y ambiguos**

Cuando se habla de requerimientos deficientes o ambiguos se trata de lo que los clientes no saben expresar y por lo tanto es difícil que el quipo de desarrollo logre entender en su totalidad lo que las personas quieren, haciendo que esta tarea se vuelva más extensa o larga de lo que podía ser al no entender bien su objetivo.

Como equipo de desarrollo se debe tener la idea clara debido a que se tiene que saber que tiene que hacer el sistema para que haga lo que el cliente quiere, algunas de las recomendaciones para que esto no suceda son:

- Ser claros con el objetivo del requerimiento
- Consultar dudas al momento de la toma de requerimiento para que todo quede por escrito y entendido en su totalidad.
- No aceptar requerimientos muy generales, debido a que la generalidad puede causar ambigüedad y malentendidos en el desarrollo.

- Recibir los requerimientos por escrito, con todo lo necesario detallado y si es necesario tener reuniones para entenderlo.

Por otro lado, para tomar buenos requerimientos se deben seguir algunas de las siguientes recomendaciones:

- Planear una línea argumentada y especificar los temas del requerimiento.
- Colocar límites de tiempo acorde a una entrega con resultados funcionales y no dar tiempos demasiado cortos donde se tienen que ampliarlos en el camino.
- Sugerir desarrollos y formas de resolver el requerimiento de una manera que talvez el cliente no a considerado, y hacer saber de los riesgos que existen al realizarlos de otra manera.
- Establecer contacto continuo entre ambas partes para saber que el requerimiento no se a salido de la línea principal de su objetivo.
- Pensar a futuro en como repercuten las desiciones tomadas en los requerimientos actuales, para no caer en requerimientos futuros que contradigan al requerimiento anterior.
- Analizar si el requerimiento es aplicable o no.

## **2.2. Cálculos ineficientes en tiempos de desarrollo**

Otro problema que se da a la hora de realizar un proyecto de IT es que a veces los tiempos son muy optimistas y no se toman en cuenta los riesgos que dicho desarrollo puede llevar a cabo, algunos de ellos pueden ser:

- la verificación de alguna licencia, esto puede llevar tiempo extra ya que depende de empresas ajenas al desarrollo y se debe acomodar a su disponibilidad.
- Detección de un software de tercero que no de respuesta, en estos casos se depende del soporte de la empresa que esta brindando la herramienta que debería responder para terminar el flujo.
- Atrasos en los requerimientos o cambios inesperados esto atrasa el proyecto, esto es común que se de cuando existe una mala comunicación entre ambas partes (el cliente y el desarrollador).
- Temas de infraestructura, esto debido a que algún servidor requiera de aprobaciones o bien los equipos sufran de un desperfecto y se pierda tiempo en lo que se reestablecen para volver a ser útiles.

Partiendo de estos riesgos se debe ser cuidadosos para calcular los tiempos, se debe de tomar en cuenta que cualquier cosa puede pasar y se debe tener tiempo para poder resolver estos errores y dar un proyecto entregado en tiempo.

### 2.2.1. Implementacion de Planning Poker

El planning Poker es una técnica donde se estimarán los elementos necesarios en el desarrollo de software utilizando metodologías ágiles. Esta técnica permite medir elementos del producto backlog, es decir la cantidad de trabajo que conlleva la realización de cada tarea, subtarea y elemento específico del proyecto de software.

#### 2.2.1.1. Escala de estimación

La escala de estimación más utilizada en el Planning Poker es que fue propuesta por Mike Cohn y se basa en la secuencia de numeros Fibonacci.

Tabla I. Escapa Planning Poker

Escala para Planning Poker								
					3	0	0	00

Fuente: elaboración propia, realizado con Word.

En el momento que se utiliza esta técnica, las tareas del Product Backlog son agrupadas por tamaño y a cada grupo se le asigna el mismo número en la escala mencionada anteriormente.

#### 2.2.1.2. Reglas del Planning Poker

- Todo el equipo de trabajo participa en la sesión. Cada integrante recibe un grupo de cartas de Planning Poker.

- El Product Owner presenta la tarea del Product Backlog que se van a estimar.
- El equipo de desarrollo delivera acerca de la tarea a estimar y realiza preguntas al Product Owner que deberan ser respondidas para realizar mejor la estimacion.
- Cada uno de los participantes selecciona una carta que representará su estimado.
- Se muestran simultaneamente las cartas seleccionadas por cada participante.
- Si todos los participantes han seleccionado la misma carta, se abrá logrado el consenso. El resultado se le asigna como estimación a la tarea de Product Backlog.
- Si los participantes no eligieron la misma carta, se realiza una discusión con el objetivo de exponer las premisas tomadas en cada estimación, dudas de la tarea o respuestas obtenidas, todos los participantes deberan justificar su eleccion de carta y en base a eso se tomará una decisión.
- Luego de la discusión se regresa al paso 3. El ciclo se repetirá hasta lograr clasificar todas las tareas.

Figura 4. **Cartas de Planning Poker**

<b>1</b>	<b>2</b>	<b>3</b>	<b>5</b>
<b>8</b>	<b>13</b>	<b>20</b>	<b>40</b>
	<b>100</b>		

Fuente: elaboración propia, realizado con Word.

### **2.3. Falta de implementación de metodologías ágiles de desarrollo o implementaciones incorrectas**

Es necesario que las empresas manejen metodologías ágiles para poder tener una implementación correcta de todas las necesidades y llevar un correcto orden del desarrollo de las actividades a realizarse dentro de la empresa.

Muchas empresas tienen el problema de que carecen de esta cultura de implementar metodologías ágiles por lo cual tienen una deficiencia a la hora de realizar la maquetación y el cálculo del tiempo que se llevará a realizar estas

tareas. En seguida se hablará sobre algunas metodologías y como implementarlas para poder tener una cultura agradable dentro del desarrollo.

## **2.4. Metodologías ágiles**

Desde hace algún tiempo en el mundo del desarrollo de software han surgido muchas notaciones en el modelado que permiten una mejor visual de lo que implica este desarrollo es decir del ciclo de vida que tendrá o de como será procesado y realizado.

Esto fue dado a raíz de que en esos momentos todo el concepto de Metodología Ágil había tenido una parada significativa y es por eso que se habla de que no sirve de nada el tener un esquema de notaciones y herramientas, si no se tiene en el momento una directriz para tener la aplicación de estas, por tal motivo es que en los últimos días se viene presentando un alto interés en cuanto a las metodologías de desarrollo se refiere.

### **2.4.1. Principios de metodologías ágiles**

En el manifiesto ágil se reúnen los principios que sobresalen teniendo en cuenta cada una de las metodologías que existen y es por eso que surgen cuatro postulados que se describen a continuación:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.

- La respuesta al cambio, por encima del seguimiento de un plan.

Y aunque en las metodologías tradicionales se le da mayor valor a cada uno de los que se encuentran a la derecha, en las metodologías ágiles se le da una mayor preponderancia a los elementos que se encuentran resaltados en la izquierda, debido a que se considera más eficiente realizar un desarrollo de dichas maneras.

El desarrollo ágil de software se encuentra en el marco de lo que se denomina Ingeniería de Software, y de esta manera promueve realizar las fases por medio de iteraciones en cuanto al desarrollo a lo largo de todo el ciclo de vida del proyecto. Hoy en día existen muchos métodos de desarrollo ágil, en su mayoría minimizan riesgos, desarrollando software en cortos periodos de tiempo.

Cuando se habla que promueve iteraciones en el desarrollo esto quiere indicar que cuando el software es desarrollado en una unidad de tiempo este es llamado una iteración, que en algunos de los casos puede durar varias semanas, todo esto dependiendo de qué tan grande es el desarrollo y que tantas implicaciones puedan existir en cada paso de dicha iteración.

Cada una de las iteraciones del ciclo de vida esta desagregado por varias fases, que son:

- Fase de planificación
- Fase de análisis de requisitos
- Fase de diseño
- Fase de codificación (desarrollo)
- Fase de pruebas (revisión),

- Documentación

Cuando se da por terminada una iteración, el equipo del proyecto realiza un análisis de las prioridades del mismo para de alguna manera u otra estructurar los pasos a seguir.

Figura 5. **Herramienta de planificación empresarial ágil Gartner 2018**



Fuente: Creatio. (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

## 2.5. Principales metodologías ágiles

Para empezar y dar un conocimiento general dentro de las metodologías ágiles se encuentra una gran cantidad sin embargo las más adoptadas serán las que se presentan a continuación, dado que cada una tiene una manera particular de trabajar o de llevar a cabo el proceso de desarrollo de software, teniendo en cuenta cada uno de los elementos discutidos dentro de estas metodologías, pero yendo por un camino completamente diferente para llegar al mismo destino brindándole una gran satisfacción al cliente.

Como una gran diversidad de metodologías ágiles, se logró identificar las más reconocidas, entre las que se pueden identificar:

- Scrum
- Xp (Extreme Programming)
- Aup (Agile Unified Process)
- Kanban
- Crystal Clear
- Lean Software Development

Se identificó que existen diversidad de metodologías ágiles que hoy día son trabajadas y reconocidas, sin embargo, se planteó dar una breve descripción de lo que son las metodologías ágiles más conocidas a nivel mundial y quizás las más utilizadas en el momento por cada uno de los equipos de desarrollo de software.

### 2.5.1. Scrum

Metodologías como Scrum que se establece como un marco de desarrollo de software que esta soportado en un proceso de construcción de manera iterativa o incremental ya que se va armando, dependiendo de las tareas y necesidades del desarrollo o la actividad a realizar y se asemeja a todos los ambientes de desarrollo ágil.

Dentro de esta metodología se pueden identificar tres roles como lo son ScrumMaster que es quien tiene una serie de funciones muy de la mano de las del director de proyecto, el ProductOwner que es quien representa a todos los stakeholders y el Team que está conformado por los desarrolladores, sin embargo existen otros roles que no son muy formales pero que no se dejan de lado como lo son los stakeholders donde se puede encontrar a los clientes, proveedores, entre otros involucrados en el desarrollo de software, pero también los *managers* que en la gran mayoría son las personas que definen de alguna manera cual será el ambiente para el desarrollo del software.

Dentro de esta metodología se establece un concepto que es el *sprint* que se asemeja a lo que es para otras metodologías es una iteración, término que es establecido y estándar a la hora de trabajarla, donde el *sprint* puede estar durando entre una y cuatro semanas claro que esta magnitud es dada por el equipo de desarrollo (Team).

Dentro del *sprint* se tienen en cuenta varios elementos de gran importancia como lo son:

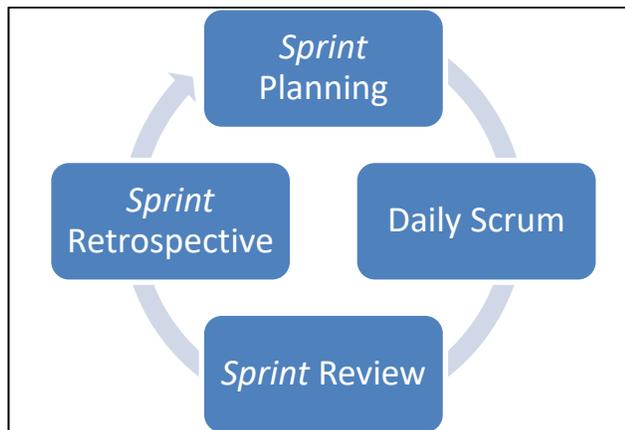
- Product Backlog: concierne a lo que describe cada uno de los requisitos priorizados.

- Sprint Planning: es la reunión donde se determinan los elementos que están en el Product Backlog y que quieren ver completados.
- Team: define cual va a ser la cantidad.
- Product Owner: es el grupo de personas que identifica dichos elementos del Product Backlog y que intervienen en el Sprint Planning, además de esto durante el transcurso del *sprint* no se puede modificar el Sprint Backlog, lo que indica que los requisitos permanecen firmes en el transcurso del *sprint*.

En esta metodología ágil predomina el concepto de Product Backlog, está compuesto por las historias de usuario y posterior a esto se realiza una priorización que será puesta en común para el desarrollo del software. Claro que no solo se tienen elementos claves como las historias de usuarios o los Product Backlog, sino que también se encuentran las Daily Scrum, que vienen a significar las reuniones que se hacen cada día del *sprint* para identificar como es la evolución del proyecto, los Scrum de Scrum, que son efectuados luego de la realización del Daily Scrum donde se discute del trabajo que se está realizando.

Además de estos se encuentra lo que se denominan las Sprint Planning Meeting que es la reunión en donde se realiza la planeación del próximo *sprint* y donde se estructurara el Sprint Backlog y demás detalles del proyecto.

Figura 6. **Flujo de SCRUM**



Fuente: elaboracion propia, realizado con Word.

### 2.5.2. **XP**

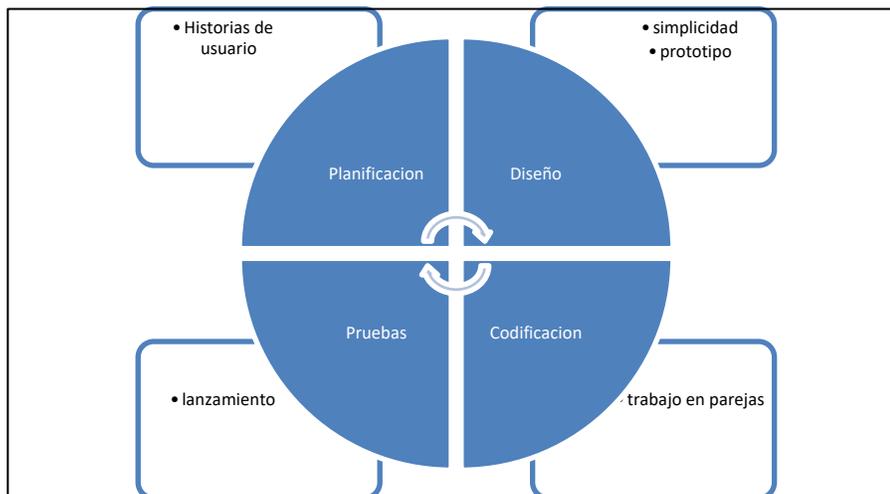
Otra de las metodologías más utilizadas y recomendadas dentro del mercado es Extreme Programming (XP), que al igual que Scrum es una de las más reconocidas a nivel mundial y más que todo es dado a apuntarle a mejorar los desarrollos de software en el ámbito de la calidad de producto y de mejorar la capacidad de apoyar al cliente atendiendo las necesidades cambiantes de los mismos.

Dentro de esta metodología las iteraciones son denominadas timeboxing y por el contrario de Scrum, en esta metodología es factible y permisible que el usuario brinde nuevos requisitos a ser adoptados.

Al igual que la metodología Scrum, tiene una serie de características que a grandes rasgos pueden verse similares pues ambas apuntan a tener presente valores como la simplicidad, comunicación, feedback, respeto entre otros,

porque son demasiado importantes a la hora de tener un buen desarrollo y desempeño del proyecto, pero su principal característica es que se enfoca más en la adaptabilidad que en lo que se denomina previsibilidad.

Figura 7. Flujo de XP



Fuente: elaboracion propia, realizado con Word.

### 2.5.3. Crystal Clear

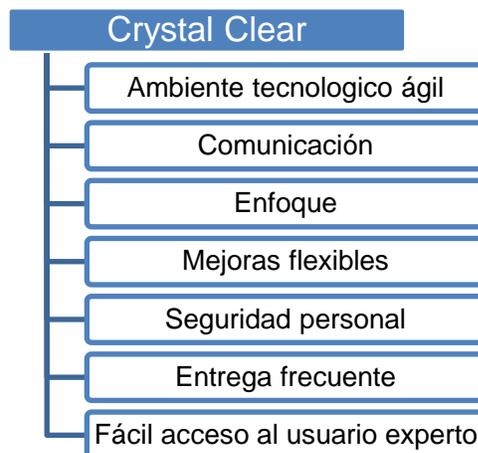
Como se sabe, otra de las principales metodologías ágiles, como Crystal Clear, al igual que las demás tienen un enfoque ágil y que además se enfoca demasiado en lo que es la comunicación pero que además es un poco más flexible con el tema de los entregables, pues en esta se manejan iteraciones que son demasiado cortas pero que presentan un feedback muy constante este permitirá ir generando correcciones tempranas en el desarrollo del software, es por eso que se implementa un rol de cliente/usuario que sea quien participe de manera activa con el tema de validaciones y verificaciones de cada uno de los requisitos funcionales, no funcionales y demás.

No solo las anteriores metodologías ágiles poseen un conjunto de valores con los que se soportan para realizar un buen trabajo, sino que Crystal también posee sus propios valores o propiedades como lo son:

- Entrega frecuente
- Comunicación osmótica
- Mejora reflexiva
- Seguridad personal
- Foco
- Fácil acceso a usuarios expertos
- Ambiente técnico con prueba automatizada
- Managment de configuración
- Integración frecuente

Cada uno de estos ayudará o apoyará de alguna manera para el desarrollo del proyecto.

Figura 8. **Propiedades Crystal Clear**



Fuente: elaboracion propia.

#### **2.5.4. Kanban**

Esta es una metodología que tiene una historia muy peculiar, dado que estas surgen a través de un sistema de producción en una de las compañías más grandes e importantes del mundo como lo es Toyota como todos saben es una de las empresas más importantes de vehículos a nivel mundial, dado que tienen presente cada uno de los principios que se plasmaron en el manifiesto ágil, dicha denominación como Kanban fue basado en las tarjetas que en aquella época fueron construidas para indicar una serie de instrucciones que permitiesen ir de lleno con el desarrollo Justo a Tiempo (Just in Time).

Esta metodología como se expreso anteriormente también tiene en cuenta cada uno de los principios de las metodologías ágiles en especial de SCRUM, pero a diferencia de esta ultima Kanban se encarga de darle un mayor protagonismo a la experimentación y a la mejora continua, es a través de esta que se puede identificar como es la visión de las diferentes perspectivas, cual es el flujo de trabajo para tomar una determinación de voy a desglosar el proyecto en diferentes componentes, para que al finalizar estar todos convergiendo para el mismo lugar. No solamente tiene esta visión, sino también que está siempre limitando el trabajo en curso o que es mejor conocido como Work in Progress y el Mide el Lead Time, que es quien se encarga de generar una optimización para que de esta manera el Lead Time tienda a ser mucho más pequeño.

A grandes rasgos el funcionamiento de esta metodología ágil está basado en entender cuál es el flujo de trabajo que se tiene con respecto a un proyecto en específico, con el fin de dividir el trabajo en bloques que van a ser plasmados sobre el esquema que presenta la figura 10 y así se esquematiza que cada uno de estos elementos se encuentre enfocado en el flujo de trabajo.

Posterior a este paso se deberá asignar el limite a cada uno de estos elementos que integran un flujo de trabajo, que como se ha mencionado anteriormente se denomina WIP (Work in Process), para posteriormente tener un tiempo promedio establecido para un ciclo del proceso, y de esta manera se optimizará cada momento el proceso y de esta manera reducir el tiempo de espera mucho más.

Figura 9. **Tablero Kanban**



Fuente: elaboracion propia, realizado con Word.

Estas son tan solo algunas de las metodologías ágiles que se pueden encontrar en el mercado del desarrollo software, pues son las más representativas y en su mayoría las más utilizadas con buenos y malos resultados en proyectos de compañías grandes y pequeñas.

## 2.6. Requisitos ágiles

Es bien sabido que los requisitos en las metodologías rígidas de desarrollo de software son un factor determinante para garantizar el éxito de los proyectos; de la misma manera juegan un papel supremo para las metodologías ágiles no obstante estas últimas no están bien clarificadas en su manera de operar e identificar los requisitos que serán el pilar para el desarrollo, por ende estas últimas no poseen unos lineamientos a seguir, que al igual que las metodologías rígidas permitan un mayor éxito en cuanto a los resultados de un proyecto.

Como se menciona anteriormente la metodología XP (Extreme Programming), fundamenta la manera de obtener los requisitos del desarrollo a través de las historias de usuarios, que a grandes rasgos no es más que lo esperan los clientes que haga el sistema, más no se centra en utilizar técnicas de elicitation de requisitos, caso contrario sucede con la aplicación de las Metodologías Rígidas en particular en la Ingeniería de Requisitos, en donde durante su etapa de planeación usa técnicas de elicitation como entrevistas, tormenta de ideas y priorización.

En la metodología Scrum las principales técnicas son los *backlogs*, los *sprints* y las reuniones diarias, para el caso concreto de los requisitos en esta metodología, los *backlogs* del producto son la manera de recolectar los requisitos mediante una lista, tal vez esta técnica pueda compararse con un documento de requisitos que contiene información necesaria para el desarrollo, esta es plasmada desde el inicio del proyecto, dicha lista va a sufrir un proceso de priorización de cada una de las tareas plasmadas allí y que posteriormente serán distribuidas en diferentes *sprints* (entendiéndose un *sprint* como un ciclo en el que se desarrollarán las tareas seleccionadas).

Una de las políticas que tiene esta metodología es que durante la ejecución del *sprint* no se podrán realizar modificaciones a los elementos allí involucrados, sino que esto podrá hacerse luego de finalizado el mismo para ser tenidas en cuenta a partir del próximo *sprint*.

Al termino de cada uno de los *sprints* se realiza una reunión de socialización con el usuario, reunión en donde se le va a mostrar al usuario lo desarrollado en este *sprint* para que de alguna u otra manera el usuario pueda dar el aval de lo que hasta ese momento se ha trabajado, aval traducido en cuanto a funcionalidad y que cumplan con cada uno de los requisitos que debería cumplir el sistema.

Dentro de esta reunión es donde el cliente puede adicionar nuevas funcionalidades o darle a conocer al equipo desarrollador nuevos requisitos a tener en cuenta en el proyecto, requisitos que deberán ser plasmados en el Backlog del proyecto y que implicará distribuirlos en los posteriores *sprints*, claro que no solo se permitiría adicionar requisitos nuevos sino también el de descartar requisitos que en dicho momento ya no sean de utilidad o necesarios para el producto final.

### **2.6.1. Principales técnicas para toma de requisitos en las metodologías ágiles**

Además de la interacción fundamental que deben tener los clientes/usuario en las metodologías ágiles e ingeniería de requisitos, como factor determinante para el éxito en el desarrollo de los proyectos existen también ciertas técnicas que se adaptan a los métodos ágiles como se ve a continuación:

- Entrevistas. Esta es la principal técnica de la ingeniería de requisitos pues es de gran conocimiento y de la más utilizada dentro de las metodologías de desarrollo ágil, pues es la que brinda un gran acercamiento con los clientes/usuarios para lograr el cometido con estas. De igual manera se utiliza en todas las metodologías ágiles, para estas la conversación directa con el cliente es el método más eficaz para elicitación u obtener la información suficiente que servirá de base o soporte para lograr los objetivos trazados y brindarle una satisfacción al cliente, de tal manera que se pueda llegar a un grado de entendimiento del problema que reduzca las inferencias o malinterpretaciones del negocio por parte del equipo de desarrollo.
- Priorización. Al igual que las entrevistas este es un método común para todas las metodologías ágiles a tal punto de que se implementan primero las características más relevantes que posee el sistema, en ciertas etapas puede ser dinámico el esquema de priorización dado que en algunos momentos pueden surgir cambios que serán tenidos en cuenta en la lista de requisitos inicial pero que como en ese inicio deben sufrir un esquema de priorización para estructurar de nuevo el esquema de trabajo a partir del momento de la inclusión de los nuevos requisitos al sistema.
- Reuniones de análisis. Se usa en algunas de las metodologías ágiles a lo largo de todo el proceso de desarrollo del producto, es decir que es transversal a este último generando una retroalimentación constante de lo que se va trabajando, técnica que consiste en sostener reuniones que permitirán realizar análisis de elementos que hayan surgido en la etapa que transcurre, cada una de estas reuniones se documentan con el fin de que puedan servir de apoyo en reuniones posteriores, también de algún

modo sirven para involucrar a todas las personas implicadas en el desarrollo proyecto.

- Modelado. Es una técnica que se utiliza en las metodologías ágiles, pero de una manera particular, pues es en estas que se utilizan con el fin de comunicar o darle a conocer de manera desagregadas cada una de las partes del sistema que se va a desarrollar, y caso contrario a como pasa en las metodologías tradicionales estos elementos no llegan a formar parte de la documentación formal.
- Documentación. Para las metodologías ágiles llevar a cabo un proceso de documentación en cada una de las fases como es llevada en las metodologías tradicionales, que son muy rigurosas y complejas, es ciertamente inviable de acuerdo a los principios que se encuentran plasmados en lo que se denomina el Manifiesto Ágil, lo que las metodologías ágiles recomiendan o en algunos casos utilizan es generar la documentación a través de los requisitos (Documento de Requisitos), en cuanto al tema de la extensibilidad se deja a consideración de cada uno de los integrantes del equipo de desarrollo o en su defecto del equipo en general, adicional a esto se tiene conocimiento que esta documentación no es muy detallado dado que se encuentra trabajando una metodología ágil
- Requisitos no funcionales. Con este tipo de requisitos, se tiene una gran dificultad en su definición cuando se está utilizando una metodología ágil, por lo tanto, no se tiene estructurado un manejo adecuado para estos. En la mayoría de los casos los clientes no los tienen muy presentes, pero van a ir surgiendo a medida que va transcurriendo el desarrollo del sistema.

### **2.6.2. Historias de usuario**

Las historias de usuario representan el tramo el cual conlleva por un camino, es decir describen cada uno de los eventos que encaminara al equipo de desarrollo a alcanzar un objetivo y que se considera son quizás la técnica más utilizada en las metodologías ágiles todo con el fin de brindarle un tratamiento a los requisitos, sin embargo, esta no es contemplada de tal forma en la Ingeniería de Requisitos tradicional.

Estas a diferencia del documento de requisitos en las metodologías tradicionales, son expresadas en lenguaje unipersonal, es decir en palabras del usuario, pero tienen el mismo propósito de los casos de uso, propósito que está enfocado a brindar una buena estimación de tiempos cuando a la planificación del proyecto se refiere, y en reemplazar el documento de requisitos.

Cuando se habla que son expresadas en lenguaje unipersonal o palabras del usuario, es dado que son los clientes quienes tienen la tarea específica de construir las historias, donde ellos plasmaran cada una de las cosas que el sistema debería hacer, excluyendo de allí toda la terminología técnica, y la principal diferencia con respecto a la Ingeniería de Requisitos es el nivel de detalle trabajado en cada una, pues es en las historias de usuario que se debe proporcionar el suficiente detalle que le permita al equipo de desarrollo obtener una estimación de los costes en la implementación de las mismas.

Es luego de esto que el equipo de desarrollo interactuara de una manera más específica con el usuario, interacción que le brindara un mayor detalle de la descripción brindada de cada uno de los requisitos plasmados en la historia de usuario, esto podría implicar de alguna u otra manera un mayor esfuerzo del que se había estimado anteriormente.

Tabla II. **Historia de usuario**

Historia de Usuario			
Numero:	1	Usuario:	Cliente
Nombre de Historia		Agregar campo de direccion	
Desarrollador Responsable		Jose Perez	
Descripcion: Agregar campo de direccion en formulario 1			
Validacion: el cliente ingresara su direccion de envio en el formulario 1			

Fuente: elaboracion propia.

## 2.7. Metodología Rup (Rational Unified Process)

Este es un proceso de la Ingeniería de Software que brinda una vista en detalle como en conjunto con la disciplina realiza la asignación de tareas y responsabilidades dentro de una organización que no necesariamente es de desarrollo.

Dicho proceso tiene como objetivo principal de brindar un aseguramiento en la construcción de software de alta calidad que de alguna u otra manera cumpla con cada una de las necesidades del cliente, teniendo presente dos aspectos importantes como lo son los costos y el calendario, siendo estos predecibles.

### 2.7.1. Características de Rup

Dentro de la metodología RUP se lograron identificar una serie de características que pueden dar claridad a lo que esta se refiere. Dichas características se describen a continuación:

- Es una metodología guiada, todo esto dado que se esta soportada sobre diagramas uml que son de gran apoyo para los usuarios finales, pues son una manera de dar claridad respecto al comportamiento del software
- Es iterativo e incremental, esto debido a que divide su proceso en cuatro fases (Iniciación, Elaboración, Construcción y Transición), con el fin de realizar una serie de iteraciones en cada una de estas, de acuerdo con el tamaño del proyecto y enfocarse en actividades que puedan ser determinantes a la hora de que resulte exitoso.
- Donde este esquema se va trabajando el desarrollo basado en componentes, para así estar bien articulados en lo que corresponde a divide y vencerás, dichos componentes serán entrelazados para generar el sistema, es por esto que se habla que esta característica permite que el sistema vaya construyéndose poco a poco mientras se desarrollan los demás componentes. Por otro lado, la utilización de un lenguaje estándar, que sea de fácil entendimiento para cualquier persona que tenga acceso a dicha información.

### **2.7.2. Elementos metodología Rup**

Cuando se habla de Metodología RUP, se deben tener presente elementos o conceptos importantes que servirán de guía para desarrollar de la mejor manera cada una de las fases que posee esta misma, por tanto, los elementos o conceptos que están ligados a esta metodología son:

- Roles: que son quienes definen el comportamiento y las responsabilidades que debe asumir una persona o un conjunto de personas, donde una persona puede estar asumiendo una diversidad de

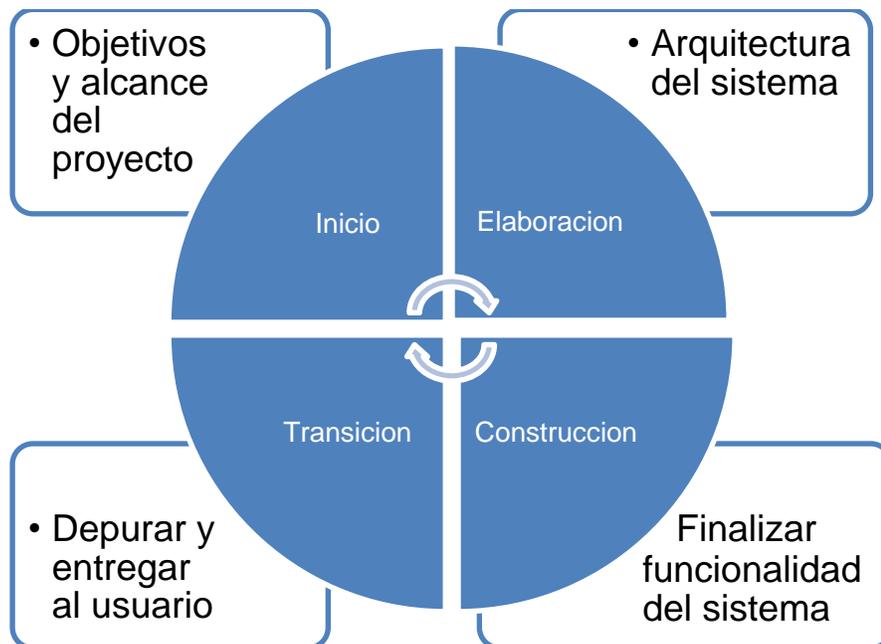
roles, a grandes rasgos son quienes dan respuesta a la pregunta ¿Quién?

- **Actividades:** estas se refieren básicamente a la acción de realizar algo específico de acuerdo a un rol asignado y serán las personas que tengan dicho rol las encargadas de realizar dichas actividades, por tanto, son estas las que den una respuesta a la pregunta ¿Cómo?, es decir el como se va a realizar o llevar a cabo la solución al problema planteado.
- **Productos:** cuando se habla de un producto se esta haciendo referencia a elementos que se consideran tangibles o que dan la certeza de un resultado del proceso que se llevo a cabo teniendo presente cada uno de los elementos que acá en este documento se plasman, elementos que estarán enfocados a irse construyendo para obtener el producto final, y son quienes darán respuesta a cuando el grupo de desarrollo se pregunta ¿Qué?, el que se va a realizar y que se debe ir construyendo para llegar a ese producto final necesario.
- **Flujos de trabajo:** cuando se habla de flujos de trabajo o *workflows*, básicamente están indicando que con realizar una definición o establecer unos roles, unas actividades y unos productos se esta listo para enfocarse en el desarrollo, pero esto es aun un poco más grueso de lo normal dado que para llegar a unos buenos resultados se debe tener en cuenta que existe una relación entre cada uno de los roles dado que cada uno de ellos estará llevando a cabo una serie de actividades que puede afectar directa o indirectamente a otro rol y es por esto que se debe establecer un esquema de navegabilidad donde esté dando a conocer cuál será el camino a seguir representado en un lujo, y serán estos los que darán respuesta a la pregunta ¿Cuándo?, es decir el

cuando entra cada uno de los roles a ejercer sus funciones y generar los productos o artefactos necesarios.

Como se menciona anteriormente, dicha metodología en estos momentos posee un esquema dividido en cuatro fases que brindaran una visualización y un desarrollo del proyecto más eficiente y un mejor manejo para cumplir con los objetivos trazados.

Figura 10. **Fases del método RUP**



Fuente: elaboracion propia, realizado con Word.

### **3. HERRAMIENTAS PARA IMPLEMENTAR LOW-CODE**

En este capítulo se conocen las principales herramientas del mercado que ofrecen la posibilidad de utilizar low-code, se aprenderá sobre ellas, y a utilizarlas para apoyarse en tareas que son repetitivas sin ayuda del control humano, se aprenderán sus ventajas y desventajas y el porque es bueno utilizarlo.

#### **3.1. Uiopath**

Se mencionará y presentara a Uiopath que es una de las lideres y mejor posicionada herramienta en la implementación de aplicaciones low-code, se mostrará la información necesaria para que las empresas del sector guatemalteco puedan tomarla en cuenta en la implementación de sus futuras automatizaciones.

##### **3.1.1. Que es uipath**

Es una herramienta de RPA que se utiliza en automatizaciones de escritorio, esta herramienta es muy utilizada para automatizar tareas repetitivas y así disminuir de la mayor manera la intervención humana en dichos procesos.

Este tipo de autimatizaciones (RPA), es un software que al ser utilizado simplifica y mejora el trabajo manual, disminuye procesos y los vuelve más exactos porque al no tener interaccion humana esto se vuelve más presiso y sin errores, Así que para definir RPA, se mencionan palabras muy importantes como precisión y perfeccion; y se refiere a *bots* que realizan tareas rutinarias,

para introducir datos, tanto en formularios web, como en cualquier aplicación de escritorio.

Es decir, este tipo de software RPA es una gran alternativa para disminuir o eliminar cargas de trabajo en las personas que cuentan con grandes cargas de trabajo en tareas de back-office como lo pueden ser las áreas de finanzas, contabilidad, recursos humanos, seguros, operaciones, entre otros. El bot será educado para entender el proceso específico y poder ejecutar automáticamente las gestiones que le han sido asignadas.

En tal sentido, han surgido desde el 2018 muchos estudios relacionados a RPA y una plataforma ha sido considerada como la mejor, la importancia UiPath RPA es que, a pesar de ser un software, reacciona con “cierta lógica” humana; pero automatizando de manera perfecta, el suministro de datos sin el característico estrés, u otro comportamiento manual que impida el rendimiento.

La compatibilidad del Robotic Process Automation con el programa Excel, e integración con Citrix (plataforma de virtualización) y SAP (firma diseñadora de productos informáticos de gestión empresarial), le dan un plus.

Es impresionante, la forma como UiPath RPA detecta e imita de manera casi instantánea, tu comportamiento en Internet. Este sistema consta de una grabadora capaz de leer y reproducir desde la pantalla de tu dispositivo, todas las acciones que ejecutes.

### 3.1.2. Beneficios de implementar RPA

Una vez se entiende que es un software de RPA y como funcionará en el futuro se entenderá de una mejor manera los beneficios que este puede proporcionar y algunos de ellos son:

- Incremento de rendimiento a nivel general porque los *robots* de software pueden trabajar las 24h sin interrupción y sin paradas incrementando el tiempo en el que una persona lo realizaria.
- Reducción de costes. 10 *robots* pueden llegar a realizar el trabajo de 100 personas y el robot no saldrá tan caro como pagarles a tantas personas tanto tiempo.
- Reducción de la carga de trabajo. Los *robots* pueden ejecutar las tareas 4 o 5 veces más rápido y sin sufrir desgaste físico o cansancio como un humano.
- Eliminación de errores. Los resultados de los *robots* son 100 % fiables.
- Incrementa el compromiso. Permite a los empleados trabajar en tareas de más valor.

Siendo la ultima la más importante para el desarrollo con las tecnologías low-code porque si una persona de contabilidad necesita realizar algún requerimiento de software para reducir algún proceso o una actividad repetitiva esta persona podrá realizarlo sin ningún problema y sin necesidad de interrumpir a un desarrollador.

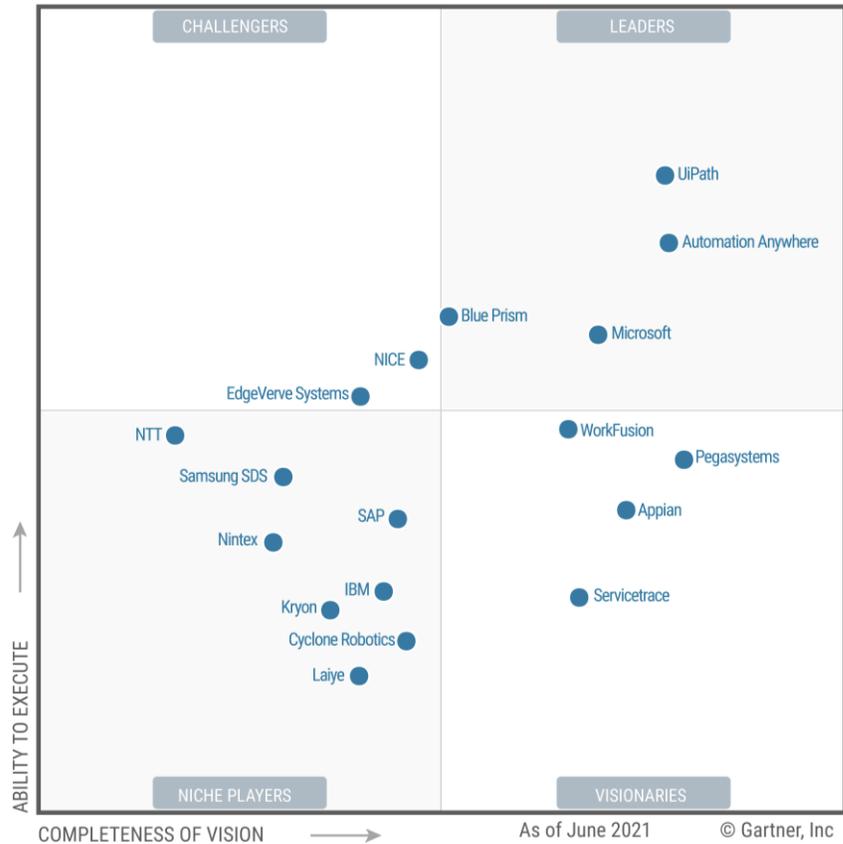
### **3.1.3. Ventajas de UiPath**

Con esta Automatización de Procesos aprovechas mejor el tiempo de tus trabajadores, al disminuir las horas del trabajo manual. Funciona de forma precisa, reproduciendo el 100 % de lo que necesitas, utilizando las tecnologías Java, Flash, HTML, Silverlight y AJAX; sin pedirte Scripts ni código alguno.

Es una plataforma de confianza y aunque su lanzamiento es reciente (2018), está en constante actualización Su diseño es funcional, diseñas los marcos a tu gusto, y personalizas tus actividades. Cuya capacidad de realizar actividades personalizadas puede ir desarrollándose desde Visual Studio de forma integral y fácil.

UiPath interactúa con sus usuarios para tomar en cuenta sus inquietudes y sugerencias; con el fin de seguir mejorando cada versión RPA.

Figura 11. RPA en el cuadrante de Gartner



Fuente: Creatio. (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

Al observar esta imagen se puede dar cuenta de que UiPath según Garner es una de los líderes en lo que es la tecnología PRA entonces se puede dar una idea que si las empresas guatemaltecas comienzan a utilizar esta herramienta pueden disminuir la carga de los desarrolladores y hacer que todos los empleados del back-office puedan implementar sus necesidades con ayuda de estos *bots*.

## **3.2. Creatio**

Se mencionará y conocerá la información necesaria para explorar la opción de Outsystems que es una de las principales herramientas para implementar low-code una vez se presente la herramienta las empresas del sector guatemalteco podrán evaluarla y tomarla en cuenta para ser utilizada en sus futuras automatizaciones.

### **3.2.1. Que es Creatio**

Es una herramienta de low-code que sirve como SaaS y su principal objetivo es la gestión de procesos y el manejo de CRM, Creatio tiene una subdivisión dentro de sus servicios, esta es:

#### **3.2.1.1. Studio Creatio**

Proyecto de Creatio orientado a la implementación y el uso de low-code, cuyo sistema utiliza una gestión inteligente para facilitar la creación de procesos empresariales por medio de soluciones y plantillas previamente cargadas.

#### **3.2.1.2. Sales Creatio**

Proyecto de Creatio que tiene como función principal la automatización de tareas para todo el flujo de ventas, orientándose en el cliente desde el momento inicial de la compra y orientándolo durante todo el proceso hasta que se realice la compra.

### **3.2.1.3. Marketing Creatio**

Proyecto de Creatio orientado a ser una plataforma de *marketing* que funciona en distintos canales para asegurar que la relación con el cliente sea eficaz y garantice ingresos a la empresa que utiliza el servicio.

### **3.2.1.4. Service Creatio**

Proyecto de Creatio que tiene como función principal la gestión de servicios inteligentes que agilizan y mejoran las soluciones para los clientes y garantizan que se cumpla lo que ellos esperan.

Una vez que se entienden las partes que conforman a Creatio, se puede definir que todo esto se une para formar una plataforma completa de low-code que se basa en el principio de que todos pueden desarrollar y realizar software para facilitar las tareas diarias.

Que quiere decir que es una plataforma de low-code como se explicó anteriormente es un software que requiere de poca o nula codificación y esto funciona para crear herramientas/aplicaciones que faciliten ciertos procesos no tan complejos.

Se sabe que Creatio a sido premiado y reconocido por analistas como Gartner, como una de las mejores tecnologías de low-code de la industria en los últimos tiempos, Creatio es conocido e implementado en más de 110 países y an demostrado que sus ventas, *marketing* y servicio a crecido exponencialmente con el uso de Creatio.

Los objetivos de Creatio en la industria son claros y directos, mejorar la efectividad y productividad de los negocios, reducir costo y tiempo en implementaciones, impulsar ventas y mejorar la experiencia del usuario.

Figura 12. **Creatio en el cuadrante de Gartner**



Fuente: Creatio. (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

### **3.2.2. Beneficios de implementar Creatio**

Una vez se entiende la amplia gama de opciones que Creatio tiene para ofrecer dentro de su tienda virtual, se pueden listar los principales y más utilizados:

- Creatio Connectos, como se sabe actualmente la comunicación a raíz del covid 19 se volvió algo esencial en todas las empresas, y se debía adaptar rápido la transformación digital y utilizar herramientas de comunicación porque no se permitía estar físicamente a todos en el mismo lugar, al ser esto uno de los principales retos como empresa muchos de los procesos que antes se hacían de una manera física pasaron a realizarse por medio de alguna red social como Whatsapp, messenger, correo electrónico o bien teams o google meet para realizar llamadas o video conferencias.

Creatio ofrece conectores predefinidos que ayudan a automatizar procesos dentro de estas redes sociales como lo pueden ser bots para una programación de llamadas o listas de difusión, o bien enviar notificaciones para cuando un proceso termine o inicie.

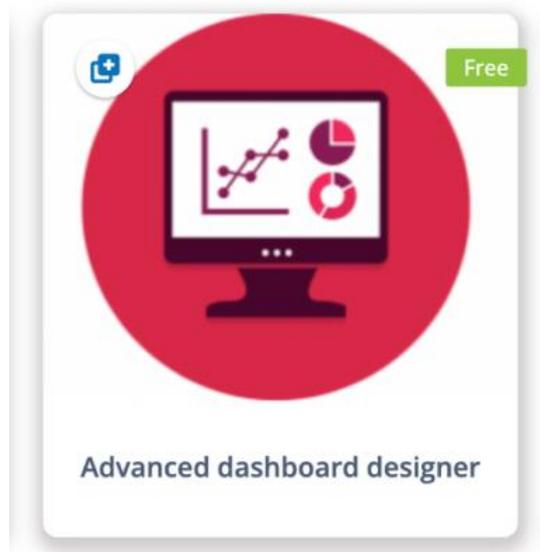
Figura 13. **Connectos de Creatio**



Fuente: Creatio. (2020). *Connectos de Creatio*. Consultado el 2 de diciembre de 2022.  
Recuperado de <https://marketplace.creatio.com/>.

- Dashboards, como se sabe actualmente el manejo de información y de datos es un activo muy importante ya que esto ayuda en la toma de decisiones y en el análisis de la información que entra y sale de las empresas, también existen desarrolladores que se encargan de realizar todo el manejo de información del lado de Data Science o Business Intelligence y estos también pueden estar cargados de trabajo por lo que Creatio ofrece una gran cantidad de herramientas que facilitarán la elaboración de *dashboard* y gráficas donde se pueden presentar datos y resultados para la gerencia y así ellos puedan tener una fácil representación de resultados y tomar decisiones, otro dato importante es que algunas implementaciones son gratis.

Figura 14. **Dashboards de Creatio**



Fuente: Creatio. (2020). *Dashboards de Creatio*. Consultado el 2 de diciembre de 2022.  
Recuperado de <https://marketplace.creatio.com/>.

### 3.3. **Ventajas de Creatio**

- Al utilizar el CRM de Creatio este brinda una herramienta única para la gestión comercial y mantiene una visión general del cliente.
- Ofrece una mejora notable en la eficiencia de los procesos comerciales, de *marketing* y de ventas dentro de la empresa que lo utiliza.
- Brinda una visión detallada en el área comercial y su avance.

- Permite segmentar el mercado de manera inteligente basandose en la informacion obtenida para priorizar los canales de ventas y los procesos comerciales.
- Permite brindar mejor información para la toma de decisiones.
- Ofrece una herramienta de low-code de codigo abierto.

### **3.4. OutSystems**

Se mencionará y explorará la información necesaria para dar a conocer a Outsystems una de las principales herramientas para implementar low-code una vez se conozca la herramienta las empresas del sector guatemalteco podrán evaluarla y tomarla en cuenta para ser implementada en sus futuras automatizaciones.

#### **3.4.1. Que es OutSystems**

Es una plataforma de Low-Code que ofrece herramientas que son útiles para que las empresas puedan desarrollar e implementar software empresarial y así facilitar sus tareas y procesos diarios garantizando una mejora de productividad y ahorro en tiempo. Es catalogada una de las mejores plataformas de Low-Code, porque brinda una interfaz amigable y fácil de entender, para que personas que se dedican a la tecnología pueden utilizarla y ayudar en el desarrollo de las aplicaciones que la empresa necesite.

Para apegarse más al entorno de desarrollo de software Outsystems brinda la opción de tener distintos ambientes tales como Desarrollo y

Produccion, también ofrece herramientas que ayudan al ciclo completo de desarrollo de las aplicaciones low-code.

Entre las actividades más utilizadas que OutSystems ofrece están:

- Creación de páginas web
- Aplicaciones móviles
- Sistemas de computo que facilitan tareas cotidianas como procesos y reportes.

Esta plataforma brinda alta compatibilidad con cualquier sistema que se desee utilizar.

Tabla III. **Compatibilidad OutSystems**

Plataforma	Compatibilidad
Web	SI
IOS	SI
Android	SI
Windows	SI

Fuente: elaboración propia, realizada con Word.

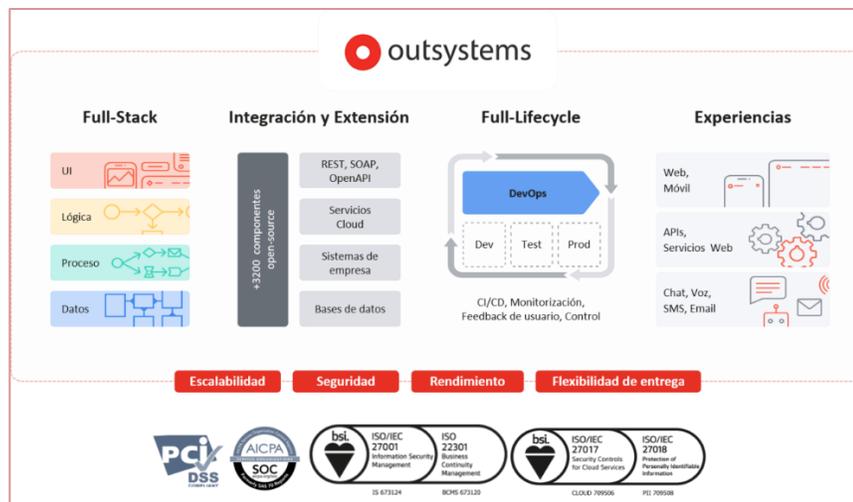
### **3.4.2. Entornos de OutSystems**

Outsystems tiene una característica muy fuerte y es que su tecnología esta en la nube es decir que sus servidores son los encargados de realizar todos los procesos, y no hay que preocuparse por el mantenimiento y soporte

de la arquitectura. Para entender mejor el flujo de trabajo de OutSystems se tienen los siguientes ambientes de trabajo.

- **Personales:** como su nombre lo indica estos son los que están unidos y ligados directamente con la persona que desarrollara la tarea/proceso, no tiene licencia su máximo almacenamiento es de 2GB y puede utilizar todas las herramientas de OutSystems exceptuando las que requieran integración con sistemas de algún propietario ajeno o softwares para control y operación de la empresa.
- **Profesionales:** en este entorno al ser un tema profesional si requiere de una licencia y todas las funciones están disponibles incluyendo las que no están disponibles para los entornos personales y el almacenamiento es configurable a la necesidad del usuario.

Figura 15. **Características OutSystems**



Fuente: low-code. (2020). *Características OutSystems*. Consultado el 11 de diciembre de 2022.

Recuperado de <https://www.altia.es/es/low-code>.

### 3.4.3. Ventajas de Outsystems

Si se utiliza Outsystems como plataforma de desarrollo de aplicaciones de low-code se tendrán muchas ventajas, pero hay que centrarse en las principales que son:

- Disminución de costo: una vez que se implementa el uso de tecnologías low-code en las empresas tiende a reducir el tiempo, por lo tanto, se obtiene de la misma manera y directamente una disminución en los costos de operación. Así mismo al utilizar herramientas low-code en el ámbito empresarial se puede disminuir el uso de desarrolladores para tareas sencillas que el mismo equipo operativo y administrativo podría realizar, dejando las tareas complejas para los desarrolladores.
- Mejores tiempos de desarrollo: cuando se utilizan herramientas de low-code como Outsystems los sistemas y las tareas se generan automáticamente, sin tener que perder tiempo en el desarrollo del código, y reducirán el tiempo de llevar la tarea al área de desarrollo.
- Reducción de soporte: cuando se crea una aplicación por medio de low-code el proceso de actualización del código y el mantenimiento con el negocio disminuye o se reduce en gran cantidad al no depender de un código para actualizar y gracias también a la disponibilidad y accesibilidad de las LCDPs.
- Compatibilidad con metodologías ágiles: las LCDPs son muy compatibles con las metodologías ágiles ya que estas ayudan a crear aplicaciones a la medida de las necesidades de los clientes o de la empresa desde un punto de vista no informático.

- Mejora en la productividad del desarrollador: al utilizar las herramientas Low-Code el personal administrativo/operativo realiza las tareas simples y así el desarrollador puede dejar a un lado esto y centrarse en el funcionamiento del software de mayor dificultad.

Figura 16. **OutSystems en el cuadrante de Gartner**



Fuente: Creatio. (2020). *Application Development Research Predicts Low-Code Tooling Takeover*. Consultado en diciembre de 2022. Recuperado de <https://adtmag.com/articles/2019/08/12/gartner-lowcode.aspx>.

Una vez estudiada una herramienta como OutSystems se pueden sacar varias conclusiones:

- Las plataformas LowCode son un adelanto para el desarrollo de aplicaciones, una única persona amplía muchísimo el trabajo. Estas herramientas tienen su propio lenguaje y tiene una serie de reglas sobre lo que se puede o no se puede hacer.
- Outsystems es una gran herramienta para el avance y desarrollo y las aplicaciones móviles.
- Finalmente, las LCDPs son el futuro debido a que su velocidad las hace fácilmente compatibles con el cambio o el desarrollo de las aplicaciones.



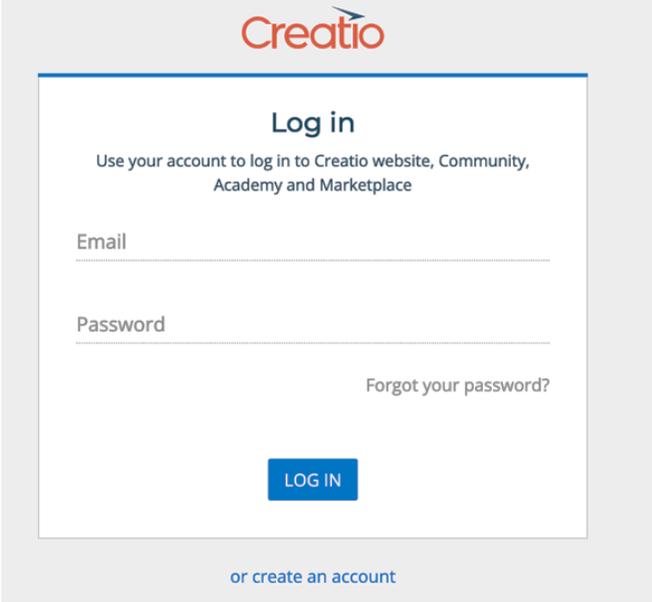
## 4. IMPLEMENTACION CON CREATIO

Una vez visto las principales herramientas para implementar las tecnologías low-code se realizará una pequeña implementación con Creatio para poder ver la facilidad con la que se utiliza.

### 4.1. Creación de la cuenta

Una vez se accede al sitio oficial de Studio Creatio se puede ingresar o bien crear una cuenta.

Figura 17. Login Creatio



The image shows a login form for Creatio. At the top is the Creatio logo. Below it is a white box with a blue border containing the text 'Log in' and 'Use your account to log in to Creatio website, Community, Academy and Marketplace'. There are two input fields: 'Email' and 'Password'. To the right of the password field is a link that says 'Forgot your password?'. Below the input fields is a blue button with the text 'LOG IN'. At the bottom of the white box is a link that says 'or create an account'.

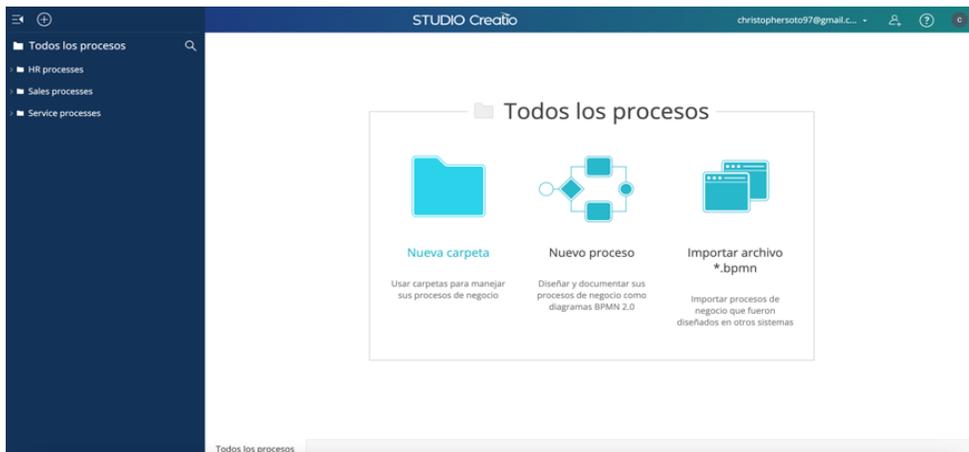
Fuente: elaboracion propia, realizado con Creatio.

## 4.2. Creación del nuevo proceso

Una vez que se ingresa a la página con las credenciales Creatio ofrece la opción de poder crear nuevos procesos de tres maneras que son:

- Creando una carpeta donde se puede almacenar un conjunto de procesos.
- Creando un proceso en individual.
- Importar algún proceso ya creado anteriormente que tenga extensión Bpmn.

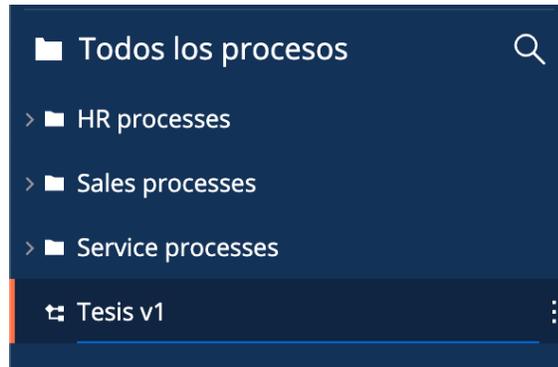
Figura 18. **Pagina inicial Nuevo Proceso Creatio**



Fuente: elaboración propia, realizado con Creatio.

Cuando se elige la opción de Nuevo Proceso se le tiene que dar un nombre al nuevo proceso para poder ubicarlo dentro de una colección de procesos.

Figura 19. **Nombrar al nuevo proceso Creatio**



Fuente: elaboracion propia, realizado con Creatio.

Como siguiente paso se observa que se tiene en la pantalla un tablero en blanco, en el se puede llenar con todas las acciones que se deseen integrar en el proceso.

Figura 20. **Proceso vacio en Creatio**



Fuente: elaboracion propia, realizado con Creatio.

#### 4.2.1. Notacion Creatio

Algunas de las tareas más comunes dentro de Creatio son las siguientes:

- Símbolo de evento de inicio: es un círculo simple que indica el primer paso de un proceso.
- Símbolo de evento intermedio: es un círculo con el borde un poco más pronunciado y representa cualquier evento que ocurre entre un evento de inicio y uno de finalización.
- Símbolo de evento de finalización: es un círculo con el borde profundo coloreado que indica el último paso en un proceso.
- Tarea: símbolo que representa cualquier tarea que será utilizada dentro del proceso.
- Tarea de usuario: tarea específica que es realizada por el usuario.
- Enviar tarea: es el encargado de compartir la información producida como resultado de un proceso.
- Recibir tarea: es el encargado de obtener la información producida como resultado de un proceso de negocio.
- Tarea manual: tarea que se realiza sin intervención de algún mecanismo o ayuda.

- Tarea de servicio: tarea que necesita un servicio externo para poder ser utilizada o realizada.
- Tarea de Script: tarea que requiere de un script para su funcionamiento.
- Tarea de regla de negocio: tarea que se basa en una regla específica de la lógica de negocio de la empresa.
- Decisión exclusiva: es un condicional el que indicará que acción tomar en base a una condición específica.
- Almacenamiento de datos: es el encargado de representa la capacidad de guardar datos asociados a un proceso o de acceder a ellos.

Figura 21. **Símbolos de evento**



Fuente: elaboracion propia, realizado con Creatio.

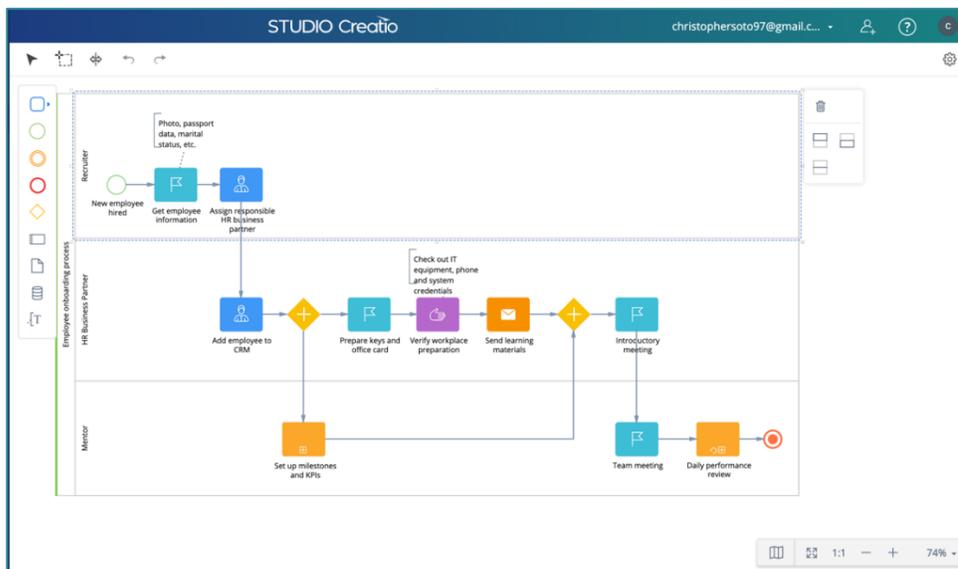
Figura 22. Símbolos de tarea



Fuente: elaboracion propia, realizado con Creatio.

Una vez que se tiene conocimiento claro de las opciones que brinda Creatio para implementar sus procesos se puede realizar un proceso que se adapte a las necesidades.

Figura 23. Proceso terminado Creatio



Fuente: elaboracion propia, realizado con Creatio.

## 5. ANALISIS DE RESULTADOS

### 5.1. Resultados obtenidos

Se presentan los resultados obtenidos en dos empresas del sector guatemalteco al utilizar herramientas low-code para mejorar su productividad y reducir sus costos.

#### 5.1.1. Empresa 1

La primera empresa para evaluar se dedica al área transporte a nivel Centro Americano la cual presentaba la problemática de querer implementar un software que les ayudará a realizar cálculos necesarios para las cotizaciones de los viajes.

A continuación, se presentarán los resultados de dicha evaluación

Figura 24. **Costo de implementación empresa 1**



Fuente elaboración propia, realizado con Word.

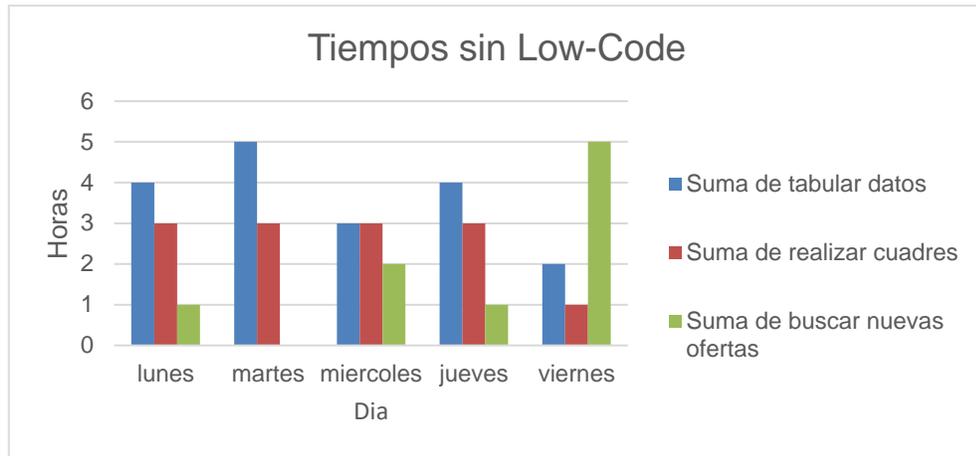
Se puede observar que el costo de contratar a un desarrollador que realizará un sitio web para que la empresa realice sus cotizaciones es demasiado elevado en comparación con pagar un curso en Udemy (<https://www.udemy.com/course/flexygo-low-code/>), para utilizar tecnologías low-code. No supera los Q230.00 y el propio dueño de la empresa puede realizar dicha tarea, y luego promocionarla entre sus colegas transportistas.

### **5.1.2. Empresa 2**

La segunda empresa por evaluar se dedica a la importación y venta de vehículos, presentaba la problemática de no tener un sistema donde pudieran llevar a cabo las facturas, contabilidad implementada en las inversiones, y ganancias de dicha compra/venta de vehículos.

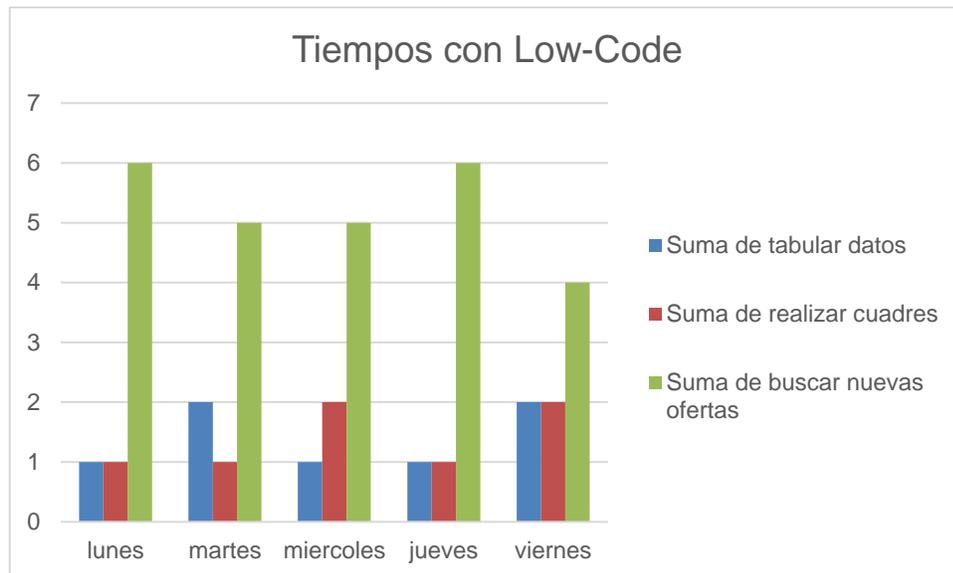
Se evaluaron los tiempos de una trabajadora en dicha venta de vehículos quien se mostraba un poco cargada la mayoría del día laboral en la tabulación de datos y de cuadros que le quitaban tiempo para poder buscar nuevas ofertas, para comprar mercadería o para estar de lleno en las ventas de dicha tienda. Invertieron en un proceso que automatizó el cuadro de los datos y que podía cargar datos sin necesidad de tabularlos uno a uno y esto mejoró los tiempos de la trabajadora, ahora puede dedicarse a otras actividades más tiempo durante el día.

Figura 25. **Tiempos sin Low-Code empresa 2**



Fuente: elaboracion propia, realizado con Word.

Figura 26. **Tiempos con Low-Code empresa 2**



Fuente: elaboracion propia, realizado con Word.



## CONCLUSIONES

1. La implementación de tecnologías low-code en las empresas guatemaltecas es totalmente viable.
2. Se mostraron comparaciones entre metodologías nuevas y errores comunes que se pueden presentar en el día a día con temas de desarrollo de software para incentivar el uso de tecnologías low-code mostrando las ventajas de estas implementaciones.
3. Se mostraron distintas herramientas, su funcionalidad, ventajas y desventajas, se explicó porque utilizar cada una de ellas para que las empresas guatemaltecas puedan elegir cual les conviene y así implementar su propuesta low-code.
4. Se comparo la productividad y costos de dos empresas guatemaltecas antes y después de haber utilizado procesos low-code el cual mostró que las tareas rutinarias disminuyeron en tiempos de procesamiento lo cual demuestra que el desempeño de la empresa ahora es más alto debido a que tienen más tiempo libre para desarrollar otras actividades y se analizó que es más barato implementar low-code que contratar más desarrolladores.



## RECOMENDACIONES

1. Segmentar las tareas de desarrollo para saber cuales son extricatamente necesarias que se hagan por desarrolladores experimentados y que tareas pueden ser elaboradas con ayuda de herramientas low-code.
2. Utilizar metodologías ágiles dentro de la elaboración de cualquier tarea o proceso dentro de la empresa, y tener una buena comunicación entre todos los interesados e involucrados dentro del proyecto, para evitar malos entendidos y situaciones comunes que pueden retrasar entregas de tareas en los proyectos.
3. Emplear herramientas que esten siempre actualizadas para poder tener una buena documentación y mantenimiento constante.
4. Evaluar costos de implementar tecnologías low-code vs desarrollos tradicionales los cuales siempre serán más costosos, así como analizar las tareas que pueden ser automatizadas para aumentar la productividad de la empresa y disminuir tiempo, costos o hasta personal que puede ser utilizado en otras áreas.



## REFERENCIAS

1. Creatio. (2020). *Creatio app elaborar proceso*. Estados Unidos: Autor.
2. Creatio Services Automation. (2020). *Servicios profesionales*. Estados Unidos: Autor.
3. Gartner. (2022). *enterprise Low-Code Application Platforms Reviews and Rating*. Estados Unidos: Autor.
4. \_\_\_\_\_. (2022). *Low – code Gartner cuadrant*. Estados Unidos: Autor.
5. Outsystems. (s.f). *Low-Code High Performance let's go*. Boston: Autor.
6. Schwaber Ken; Sutherland Jeff. (2020). *La guía definitiva de Scrum. Las reglas del juego, desarrollado y soportado por los creadores de Scrum: Español / Spanish South American*. California: Share-Alike license of Creative Commons.

