Un dibujo de una persona

Descripción generada automáticamente con confianza baja

**INSTITUTO POLITÉCNICO NACIONAL**

ESCUELA SUPERIOR DE CÓMPUTO

**Trabajo Terminal I**

“Software gestor de proyectos para el desarrollo de proyectos basados en la metodología ágil CRYSTAL”

**No. 2024-A144**

**Presentan**

Ceja Martínez Víctor Eduardo

Mirón Arévalo Gabriel Omar

Tenorio Alva Alan Osmani

**Director**

M. en C. Ocotitla Rojas Nancy

Contenido

[Capítulo 1. Introducción 2](#_Toc146030768)

[1.1 Introducción 2](#_Toc146030769)

[1.2 Planteamiento del Problema 4](#_Toc146030770)

[1.3 Propuesta de Solución 4](#_Toc146030771)

[1.4 Objetivos 5](#_Toc146030772)

[1.4.1 Objetivo General 5](#_Toc146030773)

[1.4.2 Objetivos Particulares 6](#_Toc146030774)

[1.5 Justificación 6](#_Toc146030775)

[Capítulo 2. Estado del Arte 7](#_Toc146030776)

[2.1 Tabla comparativa de softwares en el mercado y educativos 9](#_Toc146030777)

[2.2 Resultados de la comparación de Softwares 11](#_Toc146030778)

[Capítulo 3. Marco Teórico 11](#_Toc146030779)

[Software 11](#_Toc146030780)

[Metodología de desarrollo de software 11](#_Toc146030781)

[Metodología Ágil 11](#_Toc146030782)

[¿Qué es Crystal? 11](#_Toc146030783)

[Propiedades 12](#_Toc146030784)

[Roles 13](#_Toc146030785)

[FASES 14](#_Toc146030786)

[Constitución 15](#_Toc146030787)

[Ciclo de entrega 16](#_Toc146030788)

# Capítulo 1. Introducción

## Introducción

En 2001 un grupo de 17 de desarrolladores se reunieron en un retiro en Utah, para debatir acerca de los problemas a los que se enfrentaba la industria de desarrollo de software e intentar encontrar soluciones a los mismos.

Este grupo de desarrolladores detectó que uno de los principales problemas a los que se enfrentaba la industria eran las entregas prolongadas, poco ágiles y flexibles del software. La gestión de proyectos de desarrollo de software se había convertido en procesos rígidos que terminaban en productos que no atendían a las necesidades del cliente.

Así, entendieron que debían idear un proceso que permitiera la realización de cambios de forma rápida sin tener que esperar a tener el producto acabado con el objetivo de poder aportar valor al cliente en cualquier momento. El “Agile Manifesto” es el documento que resultó de este encuentro que cambiaría la forma de entender la gestión de proyectos actual [1].

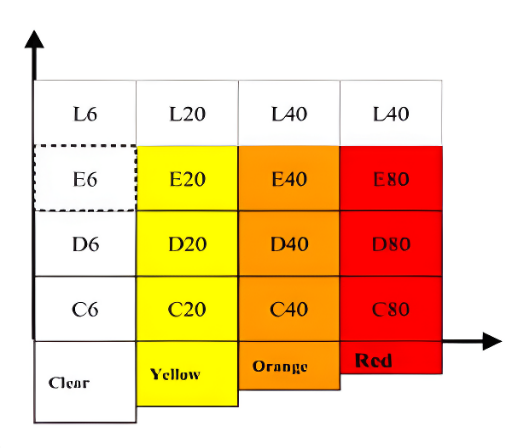
En este “manifestó” se redactan 4 valores principales(individuos e interacciones, software funcionando, colaboración con el cliente, respuesta ante el cambio) que son los pilares con los que debe contar una metodología ágil: individuos e interacciones sobre procesos y herramientas con la colaboración entre integrantes se puede lograr el mejor resultado que se pueda tener, software funcionando sobre documentación extensiva entregar al cliente el producto requerido por encima de lo prescindible para sus fines, colaboración con el cliente sobre negociación contractual permitiendo que el cliente sea quién determine la dirección que el software ha de tomar, respuesta ante el cambio sobre seguir un plan lo que ofrece la posibilidad de que los equipos cambien rápido de estrategia sin alterar el proyecto entero.

De igual forma se recitan los 12 principios del software ágil:

* Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
* Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
* Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
* Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
* Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
* El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
* El software que funciona es la medida principal de progreso.
* Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
* • La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
* La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
* Las mejores arquitecturas, requisitos y diseños emergen de equipos auto - organizados.
* A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia [2].

El proyecto se centra en la metodología ágil crystal, es una familia de metodologías con un código genético común en la cual se establecen códigos de color como parte de la definición de la complejidad de esta, si es más oscuro entonces el método es más pesado; cuánto más crítico es el sistema más rigor se necesita.

Además, crystal sugiere que se defina un color para cada proyecto en función de su criticidad y tamaño. No existe una metodología cristal en general, sino existe una metodología cristal para cada tipo de proyecto. Tal como se muestra en la siguiente figura.



Cada una de las letras que se muestran en la figura 1.1, hace referencia a los riegos potenciales:

* C: pérdida de confort debido a un fallo del sistema
* D: pérdida de dinero discrecional (nuestro dinero)
* E: pérdida de dinero esencial (este es el dinero del cual no se puede disponer)
* L: pérdida de vidas por el fallo del sistema.

Figura 1.1 Criticidad de la Metodología Crystal [3].

A su vez los números indican la cantidad de personas que son coordinadas en el proyecto, de acuerdo con lo siguiente:

• Clear es para equipos de 8 personas o menos.

• Amarillo para equipos de 10-20 personas.

• Naranja para equipos de 20-50 personas.

• Rojo para equipos de 50-100 y así sucesivamente pasando por el marrón y violeta.[3].

Como las metodologías descritas anteriormente cristal es regida por principios que hacen su utilización eficiente, entre los cuales se pueden mencionar:

* Cada proyecto necesita un grado diferente de compensación (TRADE Off).
* Entre más pequeño sea el proyecto es mejor la forma de coordinación.
* Cada uno de los proyectos necesita diferentes medios de comunicación.
* Debe existir retroalimentación y comunicación efectiva, reduciendo así los problemas en entregas fallidas.
* Los puntos dulces aceleran el desarrollo, se debe de contar con personas capaces y dedicadas a su trabajo, que se preocupen por hacer las entregas a tiempo y conforme a lo que el cliente solicita.

## Planteamiento del Problema

De acuerdo con un estudio realizado por la consultoría Auténtica para evaluar la efectividad de adoptar una metodología ágil, se encontró que más del 34% de los proyectos que utilizan metodologías ágiles fracasan, lo cual se define como proyectos cancelados o que no se entregan a tiempo [7].

La metodología crystal se caracteriza por su enfoque en la entrega temprana y funcional del software, lo cual ayuda a reducir los riesgos asociados con una prolongada etapa de desarrollo del proyecto. No obstante, uno de los desafíos que enfrenta esta metodología es la dificultad de realizar cambios en la forma en que el equipo se organizó para el desarrollo del proyecto, una vez que se ha establecido el alcance del proyecto con el cliente.

Además, el hecho de que crystal cuente con subcategorías diseñadas para equipos muy grandes puede generar confusiones al asignar tareas de forma presencial, lo cual incrementa el riesgo de error humano. Esto es especialmente problemático dado que crystal es una metodología ágil que depende de la interacción con el cliente, aunque esta interacción puede no ser tan frecuente como en otras metodologías como XP o Scrum.

Por otro lado, la flexibilidad en la asignación de roles, a pesar de ciertas restricciones, permite adaptar las responsabilidades del equipo de acuerdo con las habilidades y preferencias individuales, lo cual fomenta la motivación y el compromiso de los miembros del equipo.

La comunicación con el cliente sigue siendo fundamental en crystal, pero existen situaciones en las que el cliente no estará disponible para reuniones presenciales, lo cual puede generar retrasos en el desarrollo de las actividades hasta obtener la aprobación o el rechazo de las tareas actuales. Por tanto, resulta crucial contar con una forma organizada de visualizar las tareas y realizar un seguimiento del proyecto, a fin de evitar retrasos y asegurar el cumplimiento de los plazos establecidos.

## 1.3 Propuesta de Solución

De acuerdo con la problemática antes descrita, proponemos una solución integral que se traduce en un software de gestión de proyectos basado en los siguientes puntos clave:

* **Comunicación Integrada:** Integraremos herramientas de comunicación en tiempo real, como chat y video llamadas/llamadas, directamente en el software. En el caso de las video llamadas/llamadas, serán tomadas por un bot en caso de que el cliente no pueda tener una reunión con el equipo de desarrollo, así mismo, el bot ira tomando nota de todo lo que el cliente diga. Esto permitirá una comunicación fluida y efectiva entre los miembros del equipo y los clientes, incluso cuando la presencia física no sea posible.
* **Gestión de Roles y Responsabilidades Flexibles:** El software incluirá un sistema de gestión de roles y responsabilidades que permita la asignación flexible de tareas según las habilidades, experiencia y preferencias individuales. Esto fomentará la motivación y el compromiso del equipo al permitirles desempeñar roles que se adapten a sus fortalezas.
* **Herramientas de Planificación y Seguimiento Visual:** Integraremos herramientas visuales de planificación y seguimiento que ayuden a los equipos a organizar tareas, identificar dependencias y realizar un seguimiento en tiempo real del progreso del proyecto. Esto proporcionará una visibilidad clara y reducirá los riesgos de retrasos.
* **Registro de Decisiones y Cambios**: Implementaremos un sistema de registro de decisiones y cambios dentro del software. Esto permitirá a los equipos documentar y rastrear las decisiones tomadas a lo largo del proyecto, facilitando la gestión de cambios en la organización del equipo y la transferencia de conocimientos.
* **Generación de Reportes en Tiempo Real**: Proporcionar herramientas para generar reportes en tiempo real que muestren el estado y el progreso del proyecto. Esto permitirá a los equipos y a los “stakeholders” tomar decisiones basadas en datos actuales.

Por otra parte, debido a la magnitud que tiene el manejar proyectos grandes, solo nos enfocaremos en proyectos pequeños y medianos, por lo que solo abarcaremos de 1 a 2 colores de la metodología crystal.

## 1.4 Objetivos

A continuación, se presentarán los objetivos del desarrollo de nuestro software.

### 1.4.1 Objetivo General

Desarrollar un Software que permita planificar, gestionar y entregar un proyecto de software funcional, de alta calidad y satisfactorio para el proceso de planeación (seguimiento) bajo la metodología crystal, maximizando la eficiencia y la productividad del equipo de desarrollo.

### 1.4.2 Objetivos Particulares

* Desarrollar un módulo de creación de proyectos con obtención de características para determinar la clasificación “crystal” que sea más pertinente.
* Desarrollar un módulo de creación de tareas y /o actividades en el que de igual manera sean asignadas a uno o varios miembros del equipo.
* Implementar un videochat dentro del sistema para comunicarse con el cliente en caso de que el cliente no pueda asistir a una reunión presencial.
* Desarrollar un módulo de seguimiento del progreso de cada tarea asignada, incluyendo el tiempo dedicado a esta, el porcentaje llevado a cabo y el estado en el que se encuentra la tarea.
* Implementar un chatbot que vuelva más rápida la creación de proyectos a base de una entrada de texto.
* Desarrollar un modelo de generación de reportes para su posterior descargar en PDF en la cual se podrá visualizar el avance del proyecto en turno.

## 1.5 Justificación

El desarrollo de un software de asignación de tareas específico para proyectos basados en la metodología ágil crystal se justifica de manera sólida y respaldada tanto por la evidencia científica como por las necesidades prácticas de la gestión de proyectos de desarrollo de software.

En primer lugar, este proyecto aporta mejoras significativas en la eficiencia y productividad del equipo de desarrollo, como lo sugiere la investigación presentada en el libro "Gestión Ágil de Proyectos de Desarrollo de Software" [Smith, 2020]. La automatización de la asignación de tareas, alineada con los objetivos del proyecto, optimizará la distribución de cargas de trabajo y permitirá que el equipo se enfoque en las tareas prioritarias. Además, este enfoque promueve la colaboración y la transparencia en el proceso de asignación de tareas, dos elementos clave respaldados por la metodología crystal y la literatura académica [García et al., 2019].

La priorización de la mejora continua, ajustando el proyecto con cada entrega, como se ha observado en estudios previos [Martínez et al., 2018], asegura una progresión constante hacia el resultado final. La metodología crystal, que enfatiza la seguridad personal y el trabajo en equipo, se alinea perfectamente con este enfoque y promueve un ambiente de equipo favorable para la expresión de ideas valiosas en la elaboración del proyecto.

La metodología crystal también se destaca por su enfoque en periodos de enfoque, donde se concentra en tareas de mayor complejidad. Este enfoque, respaldado por la literatura [Cockburn, 2002], impulsa entregas puntuales y de alta calidad, características esenciales en la gestión de proyectos de desarrollo de software.

Además, el software de asignación de tareas previsto resultará útil para cualquier equipo de desarrollo que aplique metodologías ágiles similares a crystal, como XP, y ofrecerá ventajas significativas para equipos distribuidos. La automatización de la asignación de tareas mejorará la comunicación y la colaboración entre los miembros del equipo, un aspecto esencial según informes de consultoría como TechTrends [TechTrends, 2022].

En cuanto a la viabilidad del proyecto, se consideran cuidadosamente los recursos necesarios, los costos y el alcance, en concordancia con la clasificación de proyectos según su complejidad bajo la metodología crystal. Esto asegurará que el proyecto se ajuste de manera adecuada a las necesidades específicas de cada proyecto, fortaleciendo aún más su viabilidad y eficacia.

# Capítulo 2. Estado del Arte

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software/Herramienta | Descripción | Precio de la mensualidad por usuario. | Precio de la anualidad. | Tiempo de prueba gratis por usuario. |
| Monday.com | Sistema operativo de trabajo (Work OS) que permite a los equipos ejecutar proyectos y flujos de trabajo con confianza. Es un sistema operativo de trabajo simple, pero intuitivo, para que los equipos les den forma a sus flujos de trabajo, se ajusten a las necesidades cambiantes, generen transparencia, se conecten de forma colaborativa y dejen de hacer trabajo pesado manual. monday.com hace que el trabajo en equipo funcione [8]. | -Básico: MXN $160.  -Estándar: MXN $200.  -Pro: MXN $320. | Solo se encuentra disponible para Corporativos y varía el precio dependiendo de este. | Individual: Gratis de por vida. |
| Asana | Herramienta que permite gestionar el flujo de trabajo (tareas y proyectos) ofreciendo el detalle de inicio y culminación de cada una de estas, así mismo conecta a todo el equipo de trabajo permitiendo la comunicación, organización y planificación para llevar a un buen término los proyectos en cuestión [9]. | Premium: mxn $270.  Business: MXN $610. | Premium: MXN $2640.  Business: MXN $6000. | Básico: Gratis para toda la vida. |

En el presente apartado se hace una revisión exhaustiva de softwares que tratan la gestión de proyectos ya sea usando alguna metodología Ágil o no.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GoodDay Work | Plataforma de gestión de proyectos, productos y proyectos que ofrece a los equipos diferentes herramientas para la planificación, el trabajo diario y la mejora continua de todos los procesos basados en la transparencia, la responsabilidad y el reconocimiento [10]. | Profesional: USD $6 | Profesional: USD $48  Empresarial: Requiere de un presupuesto | Básico: Gratis para toda la vida |
| Spira Team | Software Scrum que respalda desde la creación de su primer proyecto hasta la gestión de la entrega de un programa grande. Las mejores prácticas están integradas, por lo que puede concentrarse en cumplir sus objetivos, sin preocuparse por lo que está cayendo a través de las grietas. Potente, directo y flexible: SpiraTeam se adapta a usted: su metodología, su flujo de trabajo, su cadena de herramientas, sus necesidades de informes [11]. | Básico: USD $60.60.  Pro: USD $31.10.  Máster: USD $18.14. | Básico: USD $2181.  Pro: USD $37,324.  Máster: USD $217,727. | No ofrece prueba o paquete gratis. |
| Zoho Projects | Aplicación de gestión de proyectos online que te permite planificar un proyecto y realizar un seguimiento completo en todo momento. Crea y asigna tareas, a ti o a otros trabajadores (ya sean de la misma empresa, colaboradores externos o clientes) [12]. | Premium: MXN $90.  Empresa: MXN $180. | Premium: MXN $72 al mes con facturación anual  Empresa: MXN $162 al mes con facturación anual | Básico: Gratis de por vida. |
| Intervals | Software personalizable de seguimiento temporal y gestión de tareas creado para las necesidades únicas de los equipos pequeños. Incluye todo lo necesario para gestionar un proyecto desde el inicio hasta la finalización. Ofrece una potente función de creación de informes de productividad [13]. | Lite: USD $29.  Basic USD $49.  Professional: USD $99 | Lite: USD $276.  Basic USD $480.  Professional: USD $960. | 30 días gratis en cualquier plan. |
| Trabajo terminal (Keisoline): Herramienta de gestión de proyectos para equipos Scrum para titulación a nivel Superior | Gestión de proyecto.  • Gestión de sprints.  • Gestión de tareas.  • Registro de tiempo.  • Gráficos de desempeño de usuario. • Cronograma de tareas [14] | Gratuito | Gratuito | Gratuito |

## 2.1 Tabla comparativa de softwares en el mercado y educativos

Con el fin de poner en evidencia la ausencia de herramientas/softwares que hagan uso de la metodología ágil crystal. A continuación, se mencionarán algunas herramientas/softwares de gestión de proyectos que han sido implementados hasta el momento.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GoodDay Work | Plataforma de gestión de proyectos, productos y proyectos que ofrece a los equipos diferentes herramientas para la planificación, el trabajo diario y la mejora continua de todos los procesos basados en la transparencia, la responsabilidad y el reconocimiento [10]. | Profesional: USD $6 | Profesional: USD $48  Empresarial: Requiere de un presupuesto | Básico: Gratis para toda la vida |
| Spira Team | Software Scrum que respalda desde la creación de su primer proyecto hasta la gestión de la entrega de un programa grande. Las mejores prácticas están integradas, por lo que puede concentrarse en cumplir sus objetivos, sin preocuparse por lo que está cayendo a través de las grietas. Potente, directo y flexible: SpiraTeam se adapta a usted: su metodología, su flujo de trabajo, su cadena de herramientas, sus necesidades de informes [11]. | Básico: USD $60.60.  Pro: USD $31.10.  Máster: USD $18.14. | Básico: USD $2181.  Pro: USD $37,324.  Máster: USD $217,727. | No ofrece prueba o paquete gratis. |
| Zoho Projects | Aplicación de gestión de proyectos online que te permite planificar un proyecto y realizar un seguimiento completo en todo momento. Crea y asigna tareas, a ti o a otros trabajadores (ya sean de la misma empresa, colaboradores externos o clientes) [12]. | Premium: MXN $90.  Empresa: MXN $180. | Premium: MXN $72 al mes con facturación anual  Empresa: MXN $162 al mes con facturación anual | Básico: Gratis de por vida. |
| Intervals | Software personalizable de seguimiento temporal y gestión de tareas creado para las necesidades únicas de los equipos pequeños. Incluye todo lo necesario para gestionar un proyecto desde el inicio hasta la finalización. Ofrece una potente función de creación de informes de productividad [13]. | Lite: USD $29.  Basic USD $49.  Professional: USD $99 | Lite: USD $276.  Basic USD $480.  Professional: USD $960. | 30 días gratis en cualquier plan. |
| Trabajo terminal (Keisoline): Herramienta de gestión de proyectos para equipos Scrum para titulación a nivel Superior | Gestión de proyecto.  • Gestión de sprints.  • Gestión de tareas.  • Registro de tiempo.  • Gráficos de desempeño de usuario. • Cronograma de tareas [14] | Gratuito | Gratuito | Gratuito |

## 2.2 Resultados de la comparación de Softwares

Como resultado de nuestra comparativa, todo los Softwares ofrecen una prueba gratuita pero solo duran un mes, dejando de lado que todos manejan como moneda de pago el dólar (USD). Nuestro software tendrá como ventaja que cobraremos en pesos mexicanos (mxn), será más barato que la competencia a excepción del proyecto educativo, pero solo está enfocado en la comunidad de ESCOM, por lo que nuestro proyecto es viable sin mencionar que es el único que usara como metodología ágil crystal.

# Capítulo 3. Marco Teórico

## Software

Es el equipamiento lógico e intangible de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

## Metodología de desarrollo de software

Es el conjunto de técnicas usadas para la elaborar proyectos que elaboran software lo dependiendo a las necesidades del proyecto en aspectos como el presupuesto del proyecto, el personal dedicado y el tiempo requerido para la entrega.

## Metodología Ágil

Son las técnicas que se centran en aplicar ciclos de trabajo recurrentes, donde se puede denotar la realización de cada etapa, entregando avances al cliente y teniendo la posibilidad de estructurar mejor los requerimientos del proyecto.

## ¿Qué es Crystal?

La metodología ágil Crystal es una colección que se centra en el desarrollo dependiendo al enfoque que necesite el proyecto, lo que distingue a crystal es la importancia de las personas y el ambiente que mantienen en la elaboración de un proyecto.

Crystal Clear es una familia de metodologías, se subdivide dependiendo a 2 factores “*Tamaño y complejidad*”, el tamaño está enfocado en la cantidad de personas asignadas al proyecto y la complejidad se basa en la exigencia del proyecto.

Para Alistar Cockburn creó la metodología según a varios proyectos de software y gracias a su propia experiencia es como se ideó que Crystal no sea basado en un proceso disciplinado, como lo es en otras metodologías, crystal clear se centra en la interacción para el cual un equipo puede lograr su objetivo a pesar de los obstáculos que se puedan enfrentar.

Al ser una metodología ágil cada vez que se realiza una iteración, se estimula al mejoramiento continuo, en el momento que se encuentre un problema, esté pueda ser corregido y la mejoría grupal avance mientras el proyecto es desarrollado.

Básicamente el equipo de desarrollo es la clave, por lo que se debe de invertir dedicación a que el ambiente sea ameno y las habilidades del equipo sean mejores, haciendo que ellos mismos creen sus propias políticas de trabajo.

## Propiedades

1. Entregas Frecuentes

Esta propiedad dicta que se debe de entregar entregas a manera de tener una funcionalidad que la pueda probar un usuario dispuesto a probar el software, para que tengamos una retroalimentación de parte del usuario.

1. Mejora Reflexiva

Se debe de aplicar diferentes técnicas para aplicar una reflexión sobre el trabajo ya hecho y como se puede mejorar, puede ser en conocimiento o en eficiencia. En Crystal la participación de cada eslabón del proyecto es importante para que el proyecto sea exitoso.

1. Comunicación Osmótica

Este término significa que la información fluye hacia el oído de fondo de los miembros del equipo, captando lo escuchado por ósmosis, esta cualidad de Crystal se realiza teniendo una comunicación efectiva en la oficina física, igualmente en los espacios virtuales puede ser una buena forma de efectuar la comunicación y la confianza del equipo. Normalmente el equipo tiene una distribución física a modo que todo el equipo pueda escuchar lo que dice un miembro de este mismo.

1. Seguridad Personal

El objetivo de crear confianza dentro del equipo hace que la eficiencia sea mejor, por lo cual esta se debe construir personalmente, disminuyendo alguna perturbación en el equipo, aclarando que los valores siempre van a limitar la interacción en el área de trabajo.

1. Enfoque

El enfoque se basa en la concentración del individuo para identificar con detalle el trabajo que realizará, así como el tiempo que puede tomarle para completarlo, aquí se denota el compromiso para hacer las tareas día a día, sin tener una preocupación como la falta de tiempo.

1. Fácil acceso a usuarios expertos

El acceso a estos usuarios proporciona un lugar para probar las entregas frecuentes e informa al equipo comentarios rápidos sobre la calidad y diseño del producto actualizados respecto a la entrega anterior.

1. Entorno técnico con pruebas automatizadas, gestión de configuración e integración frecuente

Pruebas automatizadas: Al hacer pruebas automatizadas, se asegura la calidad, que en cambio si son manuales algunos errores pueden pasar de largo.

Gestión de configuración: Permite al trabajador verificar su trabajo a parte del trabajo en conjunto de manera asíncrona, concluyendo una función en particular para que, al día del lanzamiento, vuelva a la configuración si surge un problema.

Integración frecuente: Integrar módulos es una tarea constante, por lo cual, cuanto mas frecuente se integran, cada vez más rápido se detectan errores y la eficiencia aumenta.

## Roles

* Patrocinador

Esta es la persona que asigna o defiende la asignación del dinero para el proyecto. Se supone que el Patrocinador Ejecutivo debe tener en mente la visión a largo plazo (+3 meses), equilibrando las prioridades a corto plazo con las de las versiones y los equipos posteriores que evolucionan o mantienen el sistema. Esta es la persona que creará visibilidad externa para el proyecto y proporcionará al equipo decisiones cruciales a nivel empresarial.

* Diseñador principal

Esta es la persona técnica principal, la persona que se supone que tiene experiencia en el desarrollo de software, capaz de realizar el diseño principal del sistema, puede decir cuando el equipo del proyecto está encaminado o no, y si está fuera de rumbo, cómo volver a encaminarlo. En términos de niveles de competencia. el diseñador principal es la persona con más experiencia del equipo y también se encarga de las tareas de programación más difíciles, aunque esto es común, en realidad presenta un riesgo para el proyecto, el Diseñador Líder tiende a sobrecargarse, teniendo el rol de Coordinador además de arquitecto, mentor y programador más experimentado.

* El experto empresarial

Es el experto en cómo funciona el negocio, qué estrategias o políticas están fijadas, qué es probable que varíe pronto, con frecuencia o rara vez. Esta persona responderá a todas las variadas preguntas que tendrán los desarrolladores sobre el corazón del sistema.

* Usuario embajador

Esta es la persona que se supone debe estar familiarizada con los procedimientos operativos y el sistema en uso (si ya existe uno), sabiendo cuáles son los modos de operación utilizados con frecuencia (y con poca frecuencia), qué atajos se necesitan y qué información deben estar visibles juntos en la pantalla al mismo tiempo. Esta es una base de conocimientos diferente a la que se espera que tenga el experto en negocios.

* Los diseñadores-programadores (incluido el diseñador principal)

Son responsables de los borradores de pantalla, el modelo de dominio común, los bocetos y notas de diseño, el código fuente, el código de migración, las pruebas y el sistema empaquetado.

* El Coordinador, con la ayuda del equipo,

Es responsable de producir el Mapa del Proyecto, el Plan de Lanzamiento, el Estado del Proyecto, la Lista de Riesgos, el Plan y Estado de la Iteración, y el Calendario de Visualización.

El Coordinador es responsable de dar a los patrocinadores del proyecto visibilidad sobre la estructura y el estado del proyecto, también es alguien con un buen toque humano y puede facilitar las discusiones y reducir los conflictos.

* El Probador (quien ocupe ese rol en este momento)

Es responsable de producir los Informes de error en ese momento.

* El escritor

Es responsable de producir el texto de ayuda al usuario.

## FASES

El ciclo de un proyecto en Crystal Clear tiene tres partes:

* Constitución.
* Ciclos de entrega.
* Conclusión del proyecto.

El proyecto se puede dividir de la siguiente manera, dando paso a los siguientes pasos a nivel interior: “C” Constitución, “p” Planeación de la iteración, “s” Preparación diaria, “dc” Control de desarrollo, “i” Integración, “R” taller de reflexión, “D” Entrega, “W” Conclusión.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

### Constitución

Dura unos días hasta algunas semanas, esta etapa consta de 4 partes: Construir el equipo, Realizar el Exploratory 360°, Dar forma y afinar las convenciones de la metodología y Construir el plan.

Para construir el equipo comenzamos con un patrocinador ejecutivo, el diseñador principal y el usuario experto; normalmente se añaden entre 2 y 5 personas al equipo con la capacidad de desarrollar software (Por cada 2 o 3 desarrolladores novatos debe haber 1 desarrollador experimentado).

El patrocinador ejecutivo proporciona el dinero y la dirección esencial al grupo, siempre deberá de mantener al equipo con el objetivo en la mente, así como su deber es informar al equipo sobre cualquier cambio en el negocio.

El diseñador principal es el desarrollador mas capacitado, se encarga de capacitar a los desarrolladores novatos, este interactúa con el patrocinador ejecutivo y con el usuario final. El diseñador elabora un plan de proyecto y gestiona el desarrollo del proyecto (este es el rol que puede quedar agotado antes de acabar el proyecto, se puede asignar un diseñador auxiliar).

El usuario experto proporciona un mejor ajuste al producto final, así como, una mejor organización en el equipo, con tener al usuario un par de horas cada semana, se puede obtener buena información sobre requisitos, mostrar demostraciones y obtener comentarios sobre uso.

#### Exploratory 360°

Realiza la verificación periódica de cuestiones claves: requisitos, el modelo de dominio, la tecnología a realizar, el plan de proyecto, la composición del equipo o la metodología a utilizar.

Cada uno de estos se revisa de forma detallada para detectar si el equipo previsto y la

metodología proyectada, utilizando la tecnología prevista en torno al modelo de dominio proyectado, pueden ofrecer el valor empresarial previsto con el conjunto de requisitos proyectados de acuerdo con el borrador del plan del proyecto, Exploratory 360° resulta en una serie de ajustes en la configuración del proyecto.

#### Dar forma a la metodología

Para un equipo de crystal clear puede ajustarse en un menos de una semana, ajustando los ciclos de entrega y las iteraciones, cada cierto tiempo para adaptar al equipo (Este paso son convenciones para poder adoptar al equipo la metodología).

#### Construir el plan inicial de proyecto

Hay varias formas de construir un plan de proyecto base con crystal clear:

DSDM

Implica una serie de verificaciones de los objetivos del proyecto con el patrocinador ejecutivo, seguido de la construcción de un plan de lanzamiento y una serie de plazos.

Scrum

Describe los plazos a un mes

XP

El equipo realiza tarjetas de historia para los requisitos funcionales y son clasificados en cuadros de tiempo de 3 semanas, analiza con el patrocinador ejecutivo, la frecuencia adecuada de lanzamiento para usuarios reales y luego agrupa las iteraciones en periodos de lanzamiento.

Cada uno proporciona un plan de proyecto inicial y es lo suficientemente rápido como para poder utilizarlo periódicamente para realizar un seguimiento de los cambios en la situación del proyecto y realizar un plan de proyecto actualizado.

Toda la constitución debería tardar entre unos días y una semana en completarse, a menos que el proyecto involucre una nueva tecnología que sea difícil de evaluar.

Los resultados de la actividad de constitución son:

una votación a favor o en contra del proyecto y un grupo de personas con un borrador de constitución y una idea de lo que están haciendo.

### Ciclo de entrega

Consta de 4 etapas: Recalibración del plan de lanzamiento, una serie de una o mas iteraciones, entrega a usuarios reales y el ritual de finalización.

#### Recalibración del plan de lanzamiento

Después de la primera entrega, los miembros encontrarán nueva información para incorporar el plan de proyecto, pueden seguir con el plan original o simplemente actualizarlo teniendo en cuenta los nuevos requisitos.

El patrocinador ejecutivo esta a merced de los desarrolladores en la construcción del plan del proyecto, el controla las prioridades y la estimación de cuánto tiempo llevará cada tarea.

Si siente que el proyecto está tardando demasiado tiene la opción de reemplazar el equipo, ajustar el alcance del proyecto o idear una estrategia.

#### Entrega a usuarios reales

Es una entrega frecuente a un usuario real, el costo de la primera implementación es elevado por lo que la entrega probablemente no pueda realizarse cada mes o incluso dos meses, Crystal clear puede ser de hasta 3 meses por dos motivos: los desarrolladores pueden concentrarse y tener un mejor análisis durante tres meses y hablando sobre tiempo – dinero, implica probar el sistema, escribir manuales y crear la capacitación suficiente, lo más práctico es hacerlo cada 3 meses.

#### Ritual de Finalización

La relajación es una parte esencial humana, por lo tanto, después de una entrega, es necesario un descanso, el ritual de finalización se realiza con el fin de recuperar la energía y la motivación para la siguiente entrega. En el periodo de Finalización se puede realizar los equipos y la planificación del siguiente ciclo de entrega; la reflexión debe ser en equipo, contestando las siguientes cuestiones: ¿Qué tan eficiente fue la implantación? ¿Qué acciones se pueden tomar para reducir los riesgos de despliegue y capacitación? ¿Qué opinan los usuarios del sistema? ¿Cuáles son sus puntos fuertes? ¿Y los débiles? ¿Qué nos gustaría mantener? ¿Qué nos gustaría cambiar?

Después de reflexionar sobre el ciclo realizado para evaluar el sistema, el equipo puede sentarse y observar a los usuarios, grabar a algunos usuarios y enfocarse en el análisis del sistema.

#### Ciclo de Iteración

La duración y los formatos de las iteraciones varían según los diferentes equipos, puede ser una iteración de una semana o una iteración de dos meses

Una iteración consta de tres partes:

* Planificación de la iteración
* Actividades diarias y del ciclo de integración
* Ritual de finalización

Como se ha venido mencionando la planificación de la iteración se realiza al principio de la iteración para definir los objetivos del equipo, las tareas que se realizarán en la iteración, al transcurso de la esta, se deben de realizar las actividades para posteriormente pasar al ciclo de integración.

El ritual de finalización debe contener la reflexión del equipo y un periodo de relajación para estar preparado para la siguiente iteración.

* Ciclo de integración

Depende del periodo de tiempo en el cual se esté llevando a cabo el proyecto, algunos equipos tienen una máquina independiente que ejecuta un script de compilación y prueba continuamente, otros se integran cada poco episodio de diseño, manteniéndose en estrecho contacto con las actividades de los demás y otros más se integran una vez al día o tres veces por semana. Aunque por lo general menos tiempo es mejor,

Crystal Clear no establece la duración de su uso.

### Conclusión del proceso

Se le da la conclusión después de n ciclos de entrega, dando en el ultimo ciclo, detalles de interfaz, de formato, diseño, operación, entre otros, haciendo que el producto pueda ser lanzado de manera oficial, finalizando el proceso podemos ver que desde que empezó el proyecto, hasta su finalización, las personas involucradas tienen una mejor comunicación en el equipo, confianza a nivel personal e interpersonal.

Para dicha finalización, los lideres darán clausuras de documentación, manuales y versiones.

# Capitulo 4. Análisis del sistema

## 4.1 Introducción

Para comenzar se hará un análisis de como estará compuesto el sistema, desde lo básico, para ello, en esta sección se definirá el uso del software que se usará para el desarrollo del sistema, así como las herramientas a utilizar.

Se definirán los requerimientos funcionales para determinar los aspectos primordiales del sistema, cada aspecto a cubrir se especifica en los diagramas de casos de uso y en el modelo entidad – relación.

El análisis de factibilidad para determinar la rentabilidad del proyecto a nivel técnico, económico y operacional, cada uno desglosado para cubrir todos los aspectos de cada punto, se cubrirá los riesgos al desarrollar el sistema.

## 4.2 Herramientas y software

Para empezar con este apartado se debe tener claro que es una aplicación web, para ello el lenguaje a utilizar será javascript y las bibliotecas que tiene para el desarrollo de diferentes actividades.

La herramienta que utilizaremos es un editor de texto enfocado en el desarrollo de código, en este caso es visual studio code, para pruebas de servidor la herramienta es postman, esta herramienta es capaz de realizar solicitudes a un servidor, dándonos la respuesta de este con las cabeceras que pueda contener y el contenido a tratar.

El software para desarrollar el sistema es el MERN Stack, este paquete de desarrollo se divide en las siglas MongoDB, ExpressJS, ReactJS y Nodejs, siendo bibliotecas para el desarrollo en javascript.

### 4.2.1 Visual Studio Code

Requisitos para instalar visual studio:

* Procesador: 1.6 GHz o más rápido.
* Memoria RAM: Entre 1 y 4 GB.
* Espacio en el disco duro: Entre 3 y 130 GB.
* Tarjeta de vídeo: Compatible con DirectX 9.
* Resolución de pantalla: 1024 x 768 o superior.

### 4.2.2 MongoDB

MongoDB es un manejador de bases de datos que contiene una versión gratuita, este es ideal para almacenar diferentes datos del sistema en cuestión, como lo es datos del usuario, datos del proyecto y triggers con el fin de ocasionar eventos en el proyecto.

MongoDB usado en javascript usa funciones que son bastante agradables para el uso del programador, haciendo esto que sea amena la funcionalidad de la base de datos en el sistema, sin tantas complicaciones mongodb crea bases de datos o colecciones sin que se hayan declarado previamente, así solo se tiene que preocupar por el correcto ingreso de los datos al sistema para no ocasionar inconsistencias.

Características de MongoDB:

* Alta disponibilidad con replicación y respaldo integrados
* Escalabilidad horizontal con fragmentación nativa
* Seguridad integral
* Validación nativa de documentos y exploración de esquemas con Compass (Interfaz Gráfica para el manejo de las bases)
* Herramientas de gestión de automatización, monitorización y respaldo
* Base de datos como servicio completamente elástico y con buenas prácticas integradas

### 4.2.3 ExpressJS

Express es un framework gratuito de aplicación web de Node que es mínimo y flexible que proporciona un conjunto robusto de características para aplicaciones web y móviles.

Con una gran cantidad de métodos de utilidad HTTP y middleware a su disposición, crear una API robusta es rápido y fácil.

Express proporciona una capa delgada de características fundamentales de la aplicación web, sin oscurecer Node.js características que conoce y ama.

Express.js se utiliza para una amplia gama de cosas en el ecosistema JavaScript/Node.js puedes desarrollar aplicaciones, endpoints de API, sistemas de enrutamiento y frameworks con él.

### 4.2.4 ReactJS

Es un framework de javascript gratuito que permite crear aplicaciones web y aplicaciones nativas utilizando las mismas habilidades. Se apoya en las fortalezas únicas de cada plataforma para permitir que sus interfaces se sientan bien en cada plataforma.

React también es una arquitectura. Los frameworks que lo implementan le permiten obtener datos en componentes asincrónicos que se ejecutan en el servidor o incluso durante la compilación. Lea los datos de un archivo o una base de datos y páselos a los componentes interactivos.

Permite construir interfaces de usuario a partir de piezas individuales llamadas componentes, está diseñado para permitirle combinar sin problemas componentes escritos por personas, equipos y organizaciones independientes, los componentes de React son funciones de JavaScript.

### 4.2.5 NodeJS

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor, basado en JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos.

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables del lado del servidor y de red a través de servidores privados virtuales, prácticamente ninguna función en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea,

Funciona como un único proceso, lo que significa que no crea un nuevo hilo para cada petición. Un hilo es un conjunto de instrucciones que debe realizar el servidor.

En lugar de bloquear el hilo y desperdiciar recursos de la CPU esperando una respuesta, Node.js continuará trabajando en la siguiente tarea. De esta manera, puede manejar una cantidad masiva de peticiones simultáneas.

## 4.3 Análisis de Factibilidad

En la sección presente vamos a mostrar la vialidad del proyecto, donde partiremos en el concepto técnico, económico y operacional, se abordará las especificaciones del hardware a ocupar a lo largo del desarrollo del sistema, así como los costos que tomaría el desarrollo del sistema desde el esfuerzo de los desarrolladores, hasta el costo del software y herramientas que se utilizarán.

### 4.3.1 Estudio Técnico

#### Estrategia de software

Para el desarrollo del software necesitaremos computadoras con la capacidad de correr visual studio, el navegador de preferencia, nodejs.

Para el navegador pondremos como referencia Google Chrome, sus requisitos son los siguientes:

* Procesador Pentium 4 o superior
* 100 MB de espacio libre en el disco duro
* 128 MB de RAM

Con los siguientes puntos, nos damos cuenta de que lo necesario es muy básico en una computadora hoy en día, por lo cual, aquí colocamos los detalles de los equipos que el equipo utiliza.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Marca | Modelo | Procesador | RAM | Tarjeta Gráfica | Almacenamiento |
| 1 | HP | 15-bw005la | AMD dual core A9 | 12 Gb | AMD Radeon | 1 Tb |
| 2 | - | Personalizada | AMD Ryzen 2600 | 16 Gb | Rx 6700xt | 1 Tb |
| 3 | Huawei | Personalizada | AMD Ryzen 7 | 8 Gb | Radeon Vega 8 | 512 Gb |

#### Estrategia de Software

Como se mostró anteriormente el software que se ocupará es MERN stack, pila usada en javascript, entre otras herramientas que usaremos para la comunicación del equipo, almacenamiento del código y la documentación del sistema, con todo esto las herramientas y software se divide de la siguiente manera:

* Modelado
  + Nodejs
* Codigo
  + MERN (javascript)
* Herramientas
  + Visual Studio
  + Discord
  + Github
  + Microsoft 365

#### Estrategia de Comunicaciones

Para la comunicación en equipo, se ha decidido usar la plataforma de Discord, esta plataforma ofrece servicio de videollamadas, entre otras opciones para una mejor organización del equipo. Para las revisiones se utilizará la plataforma teams.

#### Estrategia de los Recursos Humanos

Cada integrante del equipo realizará actividades de acuerdo con el diagrama propuesto previamente, aunque todos los integrantes del equipo están capacitados para realizar cualquier parte del documento sin ningún problema, cualquier inconveniente se ataca el problema entre todos.

### 4.3.2 Estudio Económico

### 4.3.3 Estudio Operacional

El equipo fue instruido por maestros y doctores de la Escuela Superior de Cómputo, cruzando por las diferentes asignaturas que hacen que sea posible la elaboración de este proyecto, haciéndonos aptos para elaborarlo.

Este proyecto impulsará el desarrollo de software primeramente a alumnos que son parte de la Escuela Superior de Cómputo con una metodología que no es popular, la interactividad de la aplicación se espera que sea de uso para los programadores.

Los costos tangibles del proyecto son casi nulos, por lo que únicamente el costo se centra en el esfuerzo del desarrollo del software, lo cual lo hace un proyecto rentable, el equipo de desarrollo se encuentra comprometido en cumplir los objetivos y entregar el sistema con el producto esperado.

## 4.4 Análisis de Riesgos

Los riesgos son parte del proyecto, aunque todo pueda estar bajo control, existen y tendrán un peso para hacer fracasar el proyecto, para ello catalogaremos los distintos riesgos de software:

* Proyecto
  + Rotación de personal: Personal con experiencia abandona el proyecto.
  + Hardware no disponible: El hardware esencial del proyecto no será entrega a tiempo.
* Negocio
  + Cambio de tecnología: Un producto competitivo se pone a venta antes que el sistema se complete.
  + Competencia del producto: La tecnología fundamental sobre la que se construirá el sistema se constituye por nueva tecnología.
* Proyecto y Producto
  + Retraso en las especificaciones: Las especificaciones de las interfaces esenciales no estarán a tiempo.
  + Subestimación del tamaño El tamaño del proyecto se ha subestimado.

Los riesgos pueden tener un impacto lo que hace que podamos contabilizar un riesgo y el impacto que puede tener en nuestro proyecto.

|  |  |  |
| --- | --- | --- |
| Categoría | Valor | Descripción |
| Muy bajo | 0% - 10% | Provoca impacto no operativo |
| Bajo | 10% - 25% | Impacta en la planeación del proyecto |
| Moderado | 25% - 50% | Incita a la exoneración del proyecto |
| Alto | 50% - 75% | Éxito del proyecto cuestionada |
| Muy alto | 75% - 100% | Fracaso del proyecto |

4.4.1 Riesgos en el Proyecto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nombre | Tipo | Porcentaje | Impacto |
| 01 | Priorizar actividades ajenas al proyecto | Proyecto  Producto | 45% | Moderado |
| 02 |  |  |  |  |
| 03 |  |  |  |  |
| 04 |  |  |  |  |
| 05 |  |  |  |  |
| 06 |  |  |  |  |

4.4.2 Estrategias de prevención y mitigación de los riesgos.

## 4.5 Requerimientos del sistema

## 4.6 Casos de uso

## 4.7 Diagrama de actividades

## 4.8 Prototipos

Referencias Bibliográficas

[1] L. Sarah. (2022, agosto 17). Agile Manifesto: la guía para entender la metodología Agile [Online]. Disponible: <https://asana.com/es/resources/agile-methodology>

[2] B. Kent, B. Mike, B. Arie, C. Alistair, C. Ward, F. Martin, G. James, H. Jim, H. Andrew, J. Ron, K. Jon, M. Brian, C. Martin, M. Steve, S. Ken, S. Jeff, T. Dave. (2001, Febrero). Manifiesto por el desarrollo Ágil [Online]. Disponible en: <https://agilemanifesto.org/iso/es/manifesto.html>