# Metodología Utilizada.

A continuación, se detallan y describen las diferentes metodologías, principios, convencionalismos y normas vigentes actualmente en el mercado de desarrollo de software, más puntualmente en el sector que maneja Odoo como ERP y tecnología sobre la cual desarrollar.

Para el desarrollo de este proyecto se deben tener en consideración los lineamientos planteados por las siguientes normas que modelan la calidad del desarrollo de software:

**Modelo de madurez de capacidades (CMM):** Que propone el nivel de madurez para medir la madurez del proceso de desarrollo dentro de una empresa.

**Modelo integrado (CMMI):** Que propone un modelo de entrenamiento para el mejoramiento y aumento de nivel del proceso de desarrollo de software dentro de una empresa.

**ISO 9001-2000:** Que propone el sistema para la administración de la calidad del software.

**ISO/IEC 15504:** Que propone el marco para la evaluación de como se ejecutan los procesos de desarrollo mas que el producto acabado en sí.

**Metodología de Desarrollo Ágil.**

Para la realización de este proyecto se deberán implementar los puntos encontrados dentro de la metodología de desarrollo ágil. En este orden se deberá incorporar lo siguiente:

Se deberá publicar resultados de manera rápida y continua de manera que se garantice la satisfacción del cliente en todo momento.

El equipo debe estar dispuesto y abierto a recibir cambios favoreciendo la flexibilidad del proceso de desarrollo. Además, el equipo deberá estar preparado para manejar adaptaciones recurrentes a los requerimientos a medida que avanza el proyecto ofreciendo así una ventaja frente a nuestra competencia. Teniendo en cuenta que el cliente nunca sabe totalmente lo que quiere.

El equipo asumirá un plazo de entrega corto producto del uso de la metodología de desarrollo ágil.

Se realizarán reuniones diarias de aproximadamente una hora de duración de modo que el equipo se mantenga sincronizado y cada desarrollador sepa lo que el resto esta trabajando contemplando lo establecido en el Manifiesto Ágil sobre daily meetings.

Para promover la sostenibilidad del proyecto, el resto de integrantes, y no solo los desarrolladores, seguirá trabajando hasta la entrega del mismo. De esta manera el proceso de desarrollo avanzara mas fluidamente ya que los desarrolladores tendrán personal calificado en el tema que responda dudas sobre temas funcionales.

Los miembros del equipo intercambiaran puntos de vistas sobre la forma en la que se esta trabajando. Para esto se ocuparán los meetings diarios. Esto para plantear maneras de mejorar el proceso utilizado actualmente. En ese orden los diferentes miembros adaptaran sus acciones en consecuencia.

Equipo hará uso de un backlog en el que se asentaran todas las tareas por realizar que han sido previamente distribuidas entre los diferentes miembros del equipo. Igualmente se reunirán todos los errores y bugs que necesiten resolverse, así como posibles improvements que irán quedando hasta el final.

Se hará uso de las historias de usuarios para asegurar que el producto final hará lo que el usuario necesita que haga. Se utilizará un lenguaje sencillo para plantear cómo funcionará un determinado feature desde el punto de vista del usuario. De esta manera se hace más fácil para los desarrolladores entrar en contexto de que es lo que programaran.

Se harán pruebas recurrentes al finalizar cada iteración antes de considerar el sprint como terminado y mostrar el avance al cliente.

Se hará uso del scrum Kanban para hacer una distribución visual e interactiva de las múltiples tareas que cada quien deberá trabajar. De esta manera se tiene una referencia visual en tiempo real de como va avanzando el proyecto. Para esto se dividirán las tareas en los renglones de: backlog, análisis de requerimientos, development, QA, delivered.

**Modelo a usar para el proceso del desarrollo de software.**

Para este proyecto se hará uso del Modelo de Cascada que es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

Dividiendo el proyecto en las siguientes etapas:

Etapas Ventajas Desventajas

1. Especificación de requisitos

2. Diseño del software

3. Construcción o Implementación del software

4. Integración

5. Pruebas (o validación)

6. Despliegue (o instalación)

7. Mantenimiento