**Archivo de configuración**

Contiene las constantes y evita que estas sean escritas directamente en el código, lo que permite que en caso de que alguna cambie su actualización sea rápida y evita modificaciones directas en el código.

**Robotic Enterprise framework**

Permite organizar el código por medio de maquinas de estado. Con esto se logra tener un mejor entendimiento del código y a su vez un mejor control sobre los errores que pueda ocurrir a lo largo del proceso. Tambien permite modularizar el código lo que evita que el se tengan flujos largos y difíciles de entender. Debido a que la comunicación entre máquinas de estados esta dada por argumentos se tiene una mejor trazabilidad de las variables con un estándar al momento de nombrar lo que evita tener argumentos sin uso

Cada Maquina de estado organiza el código según su función, por ejemplo, en la maquina de estado de Inicialización tendremos la lectura de archivos de configuración, el inicio de aplicativos a usar, lectura de información de entrada y creación de la cola de trabajo. La máquina de estado de get transacción data obtiene el próximo registro a ser procesado en la cola de trabajo y asi sucesivamente cada máquina de estado esta diseñada según la función que vaya a cumplir.

PRUEBAS

Para la parte de pruebas como garantía del buen funcionamiento del desarrollo se realizaran en conjunto con el equipo de negocio el cual estará constituido por 2 días de ejecuciones seguidas sin reporte de fallas, en estos días estaremos reunidos con el equipo de negocio y el equipo de desarrollo realizando ejecuciones monitoreadas en el cual en cada ejecución el equipo de negocio dará la respectiva observación de si el bot cumplió o no con la ejecución, en caso negativo se procederá a realizar los respectivos ajustes del proceso y se reiniciara los días estipulados de pruebas exitosas, los ajustes solicitados deberán estar dentro del alcance inicial del proceso estipulado en el PDD.

Estabilización

Después de cumplido el cronograma se hará un seguimiento de 2 semanas las cuales se denominaran estabilización del proceso, en las cuales en caso de falla se responderá de forma inmediata sin la necesidad de crear ticket al equipo de TI, esta respuesta estará enfocada a la solución del evento que provoco el error y no al cambio de las reglas de negocio del proceso, para esto último en caso de que se requiera es necesario establecer un control de cambios el cual estará estimado en tiempo de desarrollo aprobado tanto por el equipo de negocio como por el jefe del equipo de TI.

CLICLO DEL SOFTWARE

**FASE PLANIFICACIÓN:**

En esta primera fase se realiza el planteamiento del problema, se definen alcances y objetivos del software. Objetivos: estudio de viabilidad, realizar planificación detallada.

**FASE ANÁLISIS (definición de requisitos):**

Esta fase busca definir los requisitos que son los que dirigirán el desarrollo del proyecto de software. Objetivos: conocer los requisitos, asegurar que los requisitos son alcanzables, formalizar acuerdo con el cliente.

**FASE DISEÑO:**

En esta fase se estudian posibles opciones de implementación para el software que hay que construir, estructura general del mismo.

Objetivos: Identificar soluciones tecnológicas, asignar recursos materiales, proponer identificar y seleccionar, establecer métodos de validación, ajustar especificaciones

**FASE PRUEBAS:**

Esta fase busca detectar fallos cometidos en las etapas anteriores para corregirlos.

Objetivos: Realizar los ajustes necesarios para corregir posibles errores o inconsistencias.

**FASE MANTENIMIENTO:**

En esta fase se realizan tres puntos referenciados: mantenimiento correctivo, mantenimiento adaptativo y mantenimiento perfectivo.

Objetivos: Operación asegurar que el uso del proyecto es el que se pretendía, mantenimiento.