

## Desarrollo del prototipo software

Para el desarrollo del prototipo de calibración, dada la ambigüedad del problema a abordar y los requerimientos cambiantes se optó por el uso de los artefactos de la metodología ágil XP [1][2] y las prácticas de SCRUM para el ciclo de desarrollo.

Con el acompañamiento del cliente se definieron 6 historias de usuario las cuales se ejecutaron cada una por separado como una iteración representada por el siguiente ciclo.

### 1 CICLOS DE DESARROLLO

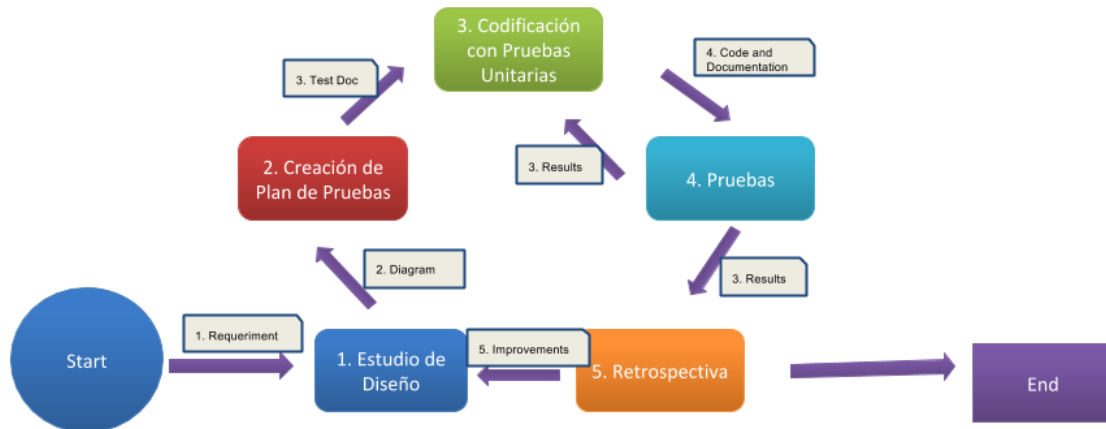


Figura 1. Ciclo de desarrollo iterativo para el desarrollo de la herramienta

### 2 HISTORIAS DE USUARIO

<b>Nombre</b>	Selección de parámetros para abstracción
<b>Numero</b>	1
<b>Tiempo estimado</b>	3 Horas
<b>Descripción</b>	Capacidad para seleccionar un conjunto de parámetros calibrables teniendo en cuenta el manual de referencia CORSIM, se debe permitir seleccionar de manera numérica: Record Type Columna inicial Columna final Valor máximo

	Valor mínimo
<b>Observaciones</b>	El usuario plantea el deseo que se haga mediante una tabla.

<b>Nombre</b>	Crear, cargar y guardar datos de salida de la simulación
<b>Numero</b>	2
<b>Tiempo estimado</b>	2 Semanas
<b>Descripción</b>	Capacidad para crear, cargar y salvar datos de salida de la simulación para netsim y fresim desde la salida del simulador, se debe tener en cuenta los datos de volumen y velocidad que se desean extraer.
<b>Observaciones</b>	Los datos deben permitir ser almacenados en archivos.

<b>Nombre</b>	Generar archivos especiales para guardar datos de campo
<b>Numero</b>	3
<b>Tiempo estimado</b>	6 horas
<b>Descripción</b>	Generar una matriz que contenga valores de volumen y velocidad vacíos, pero que contenga los enlaces con el fin de poder agregar y editar los valores medidos en campo.
<b>Observaciones</b>	Para poder tener los datos de enlace es necesario hacer una ejecución del algoritmo una vez y extraer de las salidas de la simulación los respectivos enlaces, esto cuando se desean agregar datos de campo por primera vez, pero si se agregan datos precargados de archivo no se presenta la necesidad de correr el algoritmo ya que efectivamente se cuenta con los enlaces necesarios.

<b>Nombre</b>	Extraer los parametros seleccionados para ser calibrados
---------------	--

<b>Numero</b>	4
<b>Tiempo estimado</b>	1 semana
<b>Descripción</b>	Extraer el conjunto de parámetros definidos previamente e insertar estos en un contenedor que adicionalmente guarde la información técnica de cada parámetro.
<b>Observaciones</b>	Dada la incapacidad de medir de manera directa el número de elementos a ser calibrados se debe realizar una función que permita obtener el número total de parámetros a ser calibrados. Tener en cuenta manual de referencia CORSIM Formato del contenedor y datos que deben ser usados. [0] Record type, [1] Start Column, [2] End Column, [3] Min Value, [4] Max Value, [5] Value, [6] Line Number

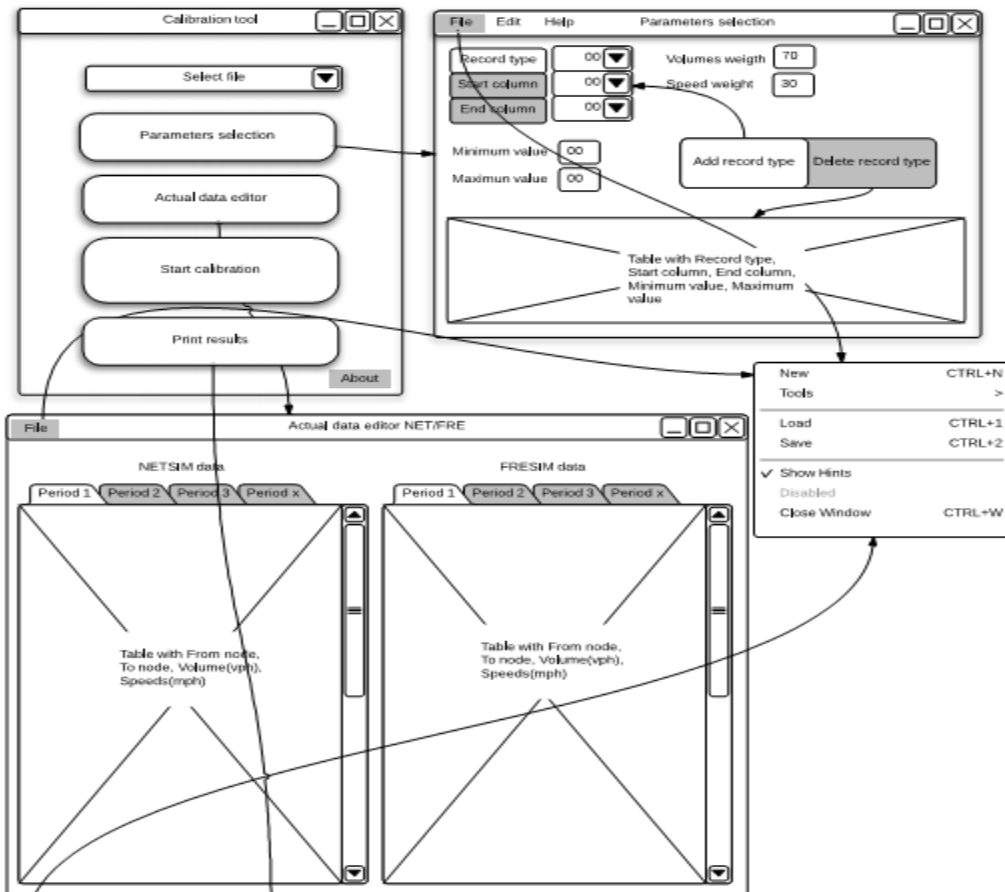
<b>Nombre</b>	Sobrescribir el conjunto de parámetros nuevamente en el TRF
<b>Numero</b>	5
<b>Tiempo estimado</b>	4 dias
<b>Descripción</b>	Dado que se posee información como el número de línea, columnas inicial a final, se debe escribir de esta manera los valores en el archivo trf después de ser calibrados o al aplicar cambios sobre ellos para realizar una evaluación del modelo.
<b>Observaciones</b>	Antes de escribir se debe validar la consistencia y restricciones de los datos para evitar errores en tiempo de ejecución del compilador

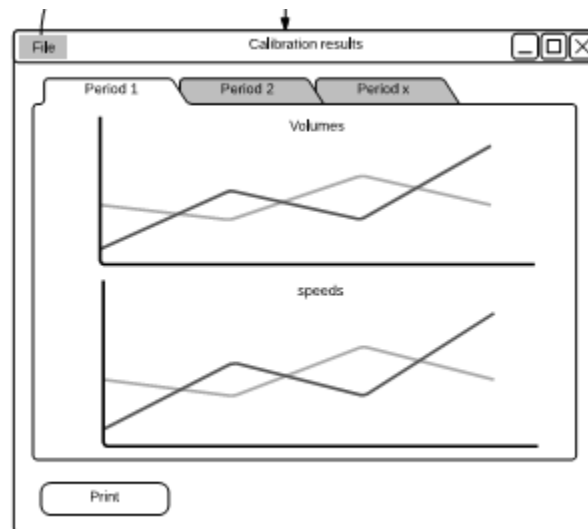
<b>Nombre</b>	Implementar algoritmo de calibración
<b>Numero</b>	6

<b>Tiempo estimado</b>	2 semanas
<b>Descripción</b>	Con el Framework para interacción con la herramienta CORSIM previamente definida se debe establecer una iteración completa para la implementación
<b>Observaciones</b>	Tener en cuenta parametrización dinámica para cada algoritmo, no deben quedar valores estáticos en el código Java.

### 3 PROTOTIPOS DE USUARIO

Las siguientes prototipos de interfaz gráfica de usuario se crearon con el acompañamiento del cliente de tal manera que la interfaz gráfica del prototipo software son tal cual lo evidencian estos prototipos.



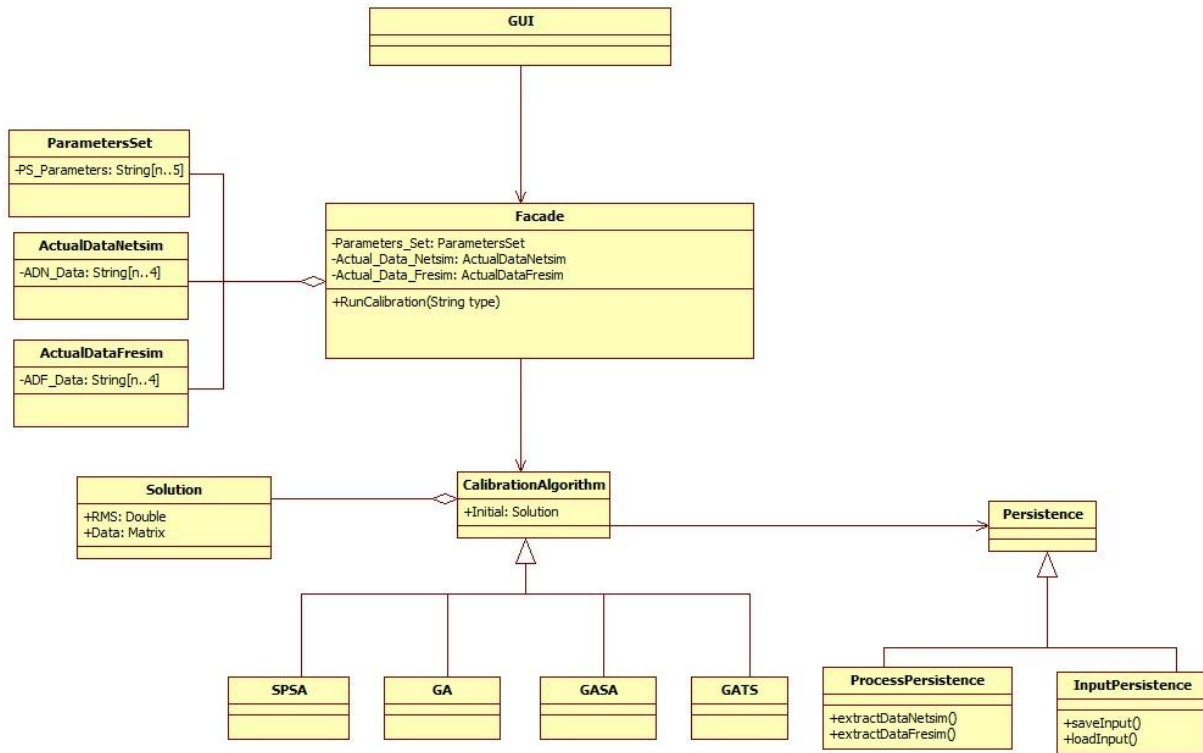


**Figura 2. Prototipos de la interfaz grafica de usuario**

#### **4 DIAGRAMA DE CLASES**

El diagrama de clases evidencia un patrón fachada el cual recibe todas las peticiones desde la interfaz de usuario y los procesa a través de las clases del dominio. Se define una clase abstracta para el algoritmo de calibración con características generales de los mismos, de esta clase abstracta hereda cada uno de los algoritmos de calibración a crear. Se puede notar como existe una clase "Solución" la cual contiene un individuo o solución al problema de calibración.

Las clase fachada contiene el conjunto de parámetros elegidos para calibrar, los valores de campo para netsim y para fresim, el algoritmo memético utiliza la persistencia basada en archivos para todos sus procesos.



## 5 REFERENCIAS

- [1] G. F. Escribano, "Introducción a Extreme Programming," *Ingeniería del Software*, vol. 1, p. 14, 2002.
- [2] D. Wells. (2001, *Extreme programming*.