

Un Marco de trabajo para la generación automática de esquemas de aprendizaje en el contexto de predicción de fallos de software mediante el uso de algoritmos genéticos

Dr. Juan Murillo-Morera

21 de mayo de 2022

Agenda

- 1 **Introducción**
- 2 **Marco teórico**
- 3 **Objetivos**
- 4 **Antecedentes**
- 5 **Metodología**
- 6 **Marco de trabajo**
- 7 **Análisis de resultados**
- 8 **Publicaciones**
- 9 **Conclusiones y trabajo futuro**

Motivación (1/2)

- Un **Fallo** es **introducido en el software** como el resultado de un **error** que puede causar una **conducta incorrecta** y no **acorde a su especificacin**. (*Glosario-IEEE, 1990*).
- Los fallos (*Singh et al., 2015*):
 - **Disminuyen la confiabilidad** y la **calidad del software**.
 - Aumenta los **costos de operación** e **insatisfacción** en los usuarios finales.
- Los **modelos de predicción**, buscan evaluar **esquemas de aprendizaje**. (*Menzies et al., 2007*)(*Song et al., 2011*).

Motivación (2/2)

- Un *esquema de aprendizaje* esta compuesto por: **técnicas de preprocesamiento, selección de atributos y algoritmos de aprendizaje** (Menzies et al., 2007)(Song et al., 2011).
- La **generación de esquemas de aprendizaje** se puede hacer mediante un **marco de trabajo**.
- Un marco de trabajo es **plataforma conceptual y concreta** donde existe un **código común** con una **funcionalidad clara**. (Gao et al., 2009).

Problema (1/2)

- Los **métodos de aprendizaje máquina** han jugado un **papel transcendental** en la construcción de **modelos de predicción** (*Singh et al., 2010*).
- Existen tres líneas de investigación en el área de fallos (*Song et al., 2011*):
 - Densidad de fallos.
 - Asociación de fallos.
 - **Propensidad a fallos.**
- En el contexto de predicción de fallos los **módulos** son las **unidades** mas **pequeña** con **funcionalidad** (*Menzies et al., 2007*).

Problema (2/2)

- No existe una **única solución** a nivel de **esquemas de aprendizaje** para todos los **repositorios de datos históricos** (*Murillo et al., 2016*).
- **Diferentes configuraciones de parámetros** dan diferentes resultados, según el dominio en estudio (*Murillo et al., 2016*).
- **Nuevas técnicas y algoritmos de aprendizaje** surgen producto de la **combinación** de las existentes (*Whitley et al., 1994*).

Justificación (1/2)

- El **incremento** del **tamaño** y de la **complejidad** de los proyectos de software es cada vez **mayor** (*Schroter et al., 2015*).
- Predecir fallos permite **ubicar** y **asignar recursos** de forma **inteligente** (*Moser et al., 2008*).
- **Aumentar** el **desempeño** en los **modelos de predicción de fallos** ayuda a **mantener** la **calidad del software**. (*Khoshgoftaar et al., 2010*).

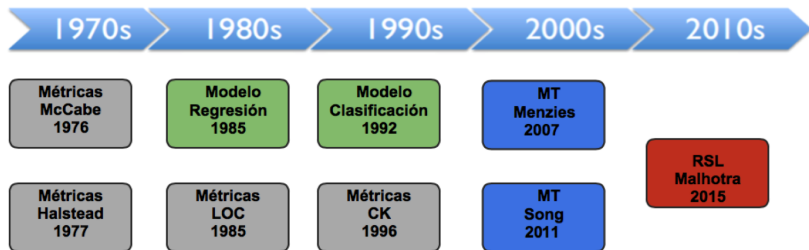
Justificación (2/2)

- Es necesario **enfocarse** en la **exactitud predictiva** de los **algoritmos evolutivos** en predicción de fallos de software (*Malhotra, 2015*).
- Un enfoque de los **algoritmos evolutivos**, son los algoritmos genéticos (*Cervigon, 2009*).
- **Automatizar** los esquemas de aprendizaje puede ayudar a obtener **mejor desempeño** (*Murillo et al., 2016*).

Pregunta de investigación

¿Cómo se puede generar de forma automática el mejor esquema de aprendizaje posible, acorde a su desempeño, tomando en cuenta las características específicas de un repositorio de datos histórico de defectos de software?

Métricas

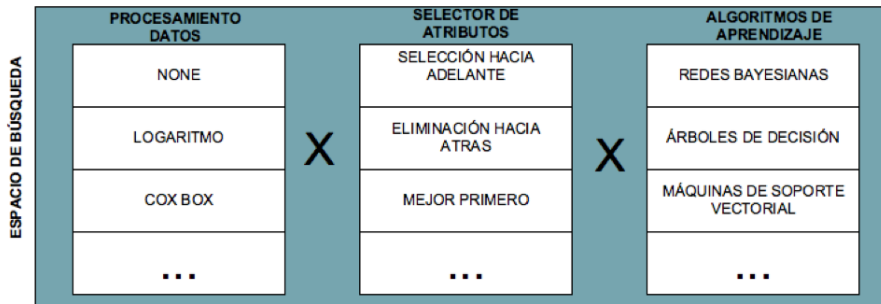


Repositorios históricos

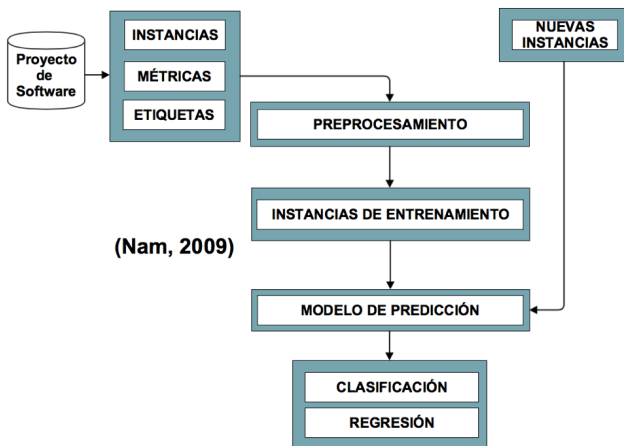
Cuadro: Estructura de un repositorio de datos histórico

	Met_1	Met_2	Met_3	Met_m	Etq
Mod_1	39	3	2	3	T
Mod_2	113	9	1	6	F
Mod_3	21	3	1	2	F
Mod_n	16	3	1	3	T

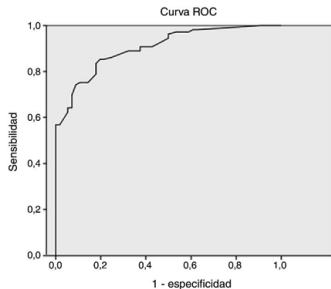
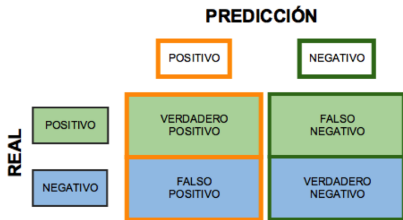
Esquemas de aprendizaje



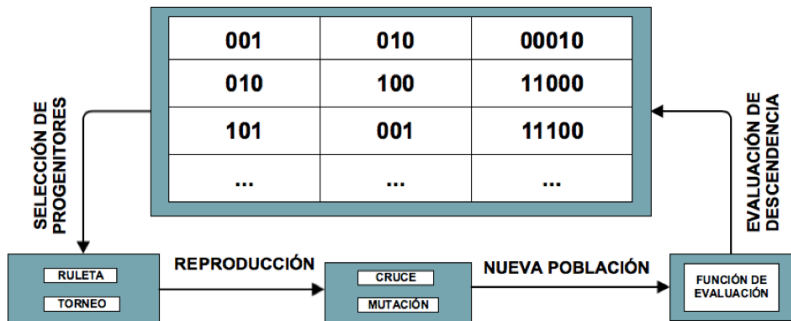
Modelos de predicción



Evaluación de los modelos de predicción



Algoritmos genéticos



General

Proponer un marco de trabajo que genere de forma automática esquemas de aprendizaje por medio de algoritmos genéticos en el contexto de predicción de fallos de software.

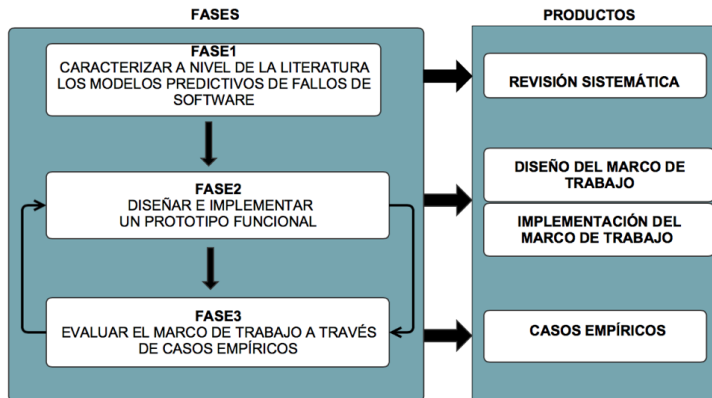
Específicos

- Caracterizar, acorde a la literatura, los modelos predictivos de fallos de software enfatizando en los esquemas de aprendizaje (técnicas de preprocesamiento, selección de atributos y algoritmos de aprendizaje).
- Diseñar e implementar el componente generador, evaluador y predictor del marco de trabajo mediante un prototipo funcional que utilice algoritmos genéticos.
- Evaluar el desempeño del marco de trabajo planteado mediante la ejecución de casos empíricos.

Marcos de trabajo existentes



Esquema

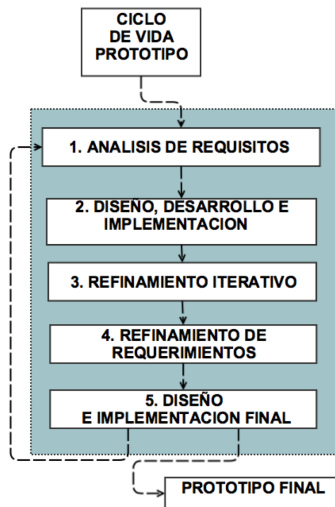


Caracterizar los modelos predictivos de fallos de software



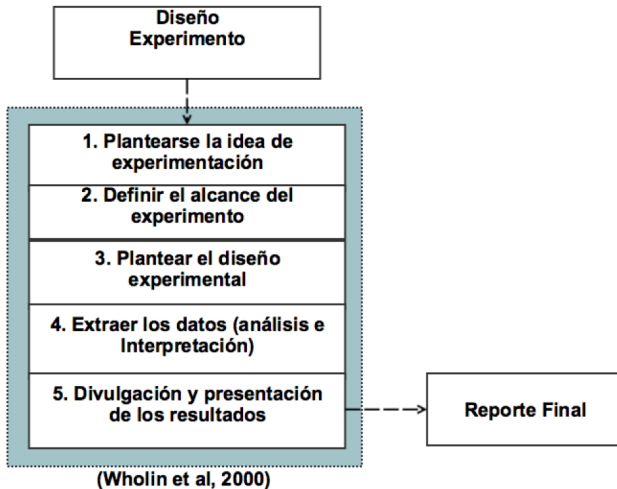
(Kitchenham et al, 2007)

Diseño e implementación de un prototipo funcional

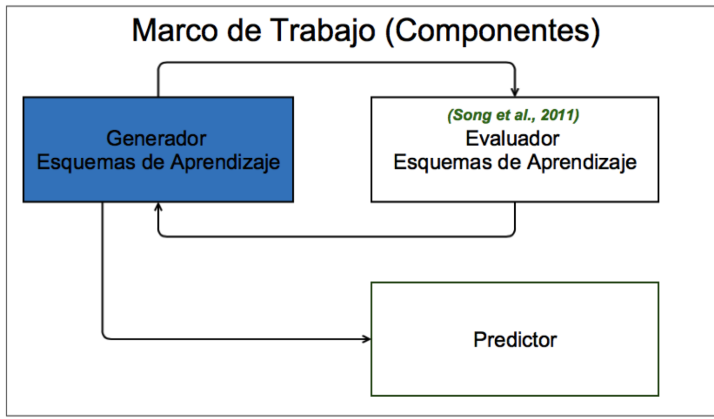


(Cataldi et al, 2009)

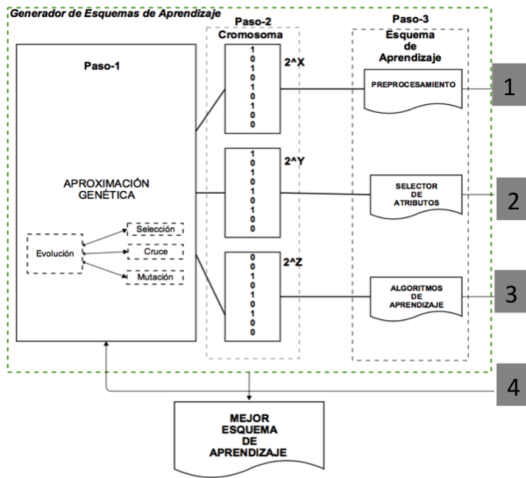
Evaluación del marco de trabajo (casos empíricos)



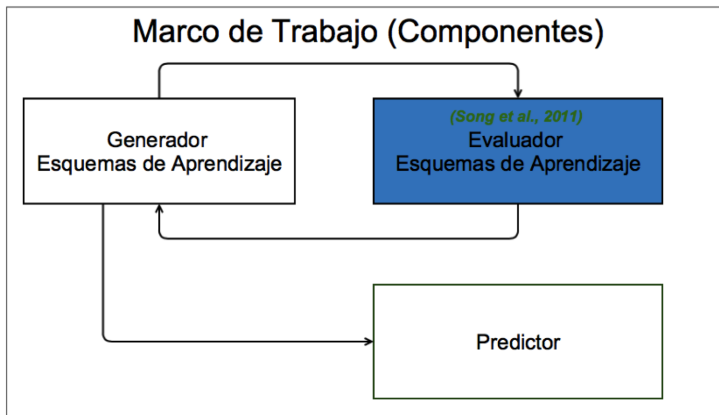
Componente generador (1/2)



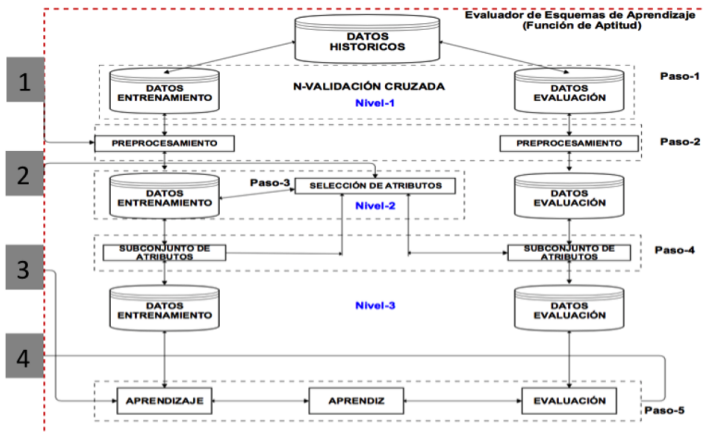
Componente generador (2/2)



Componente evaluador (1/2)

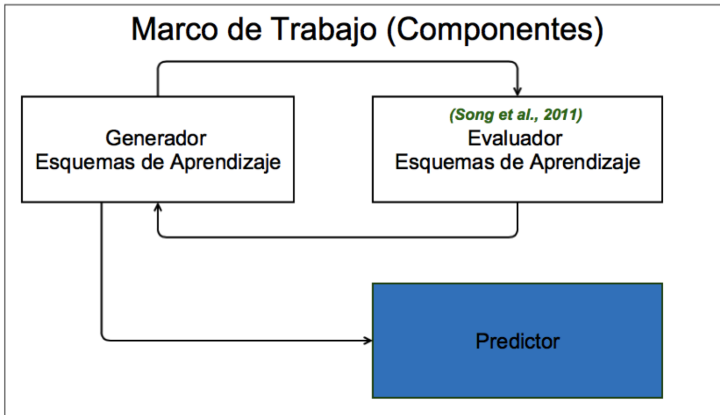


Componente evaluador (2/2)

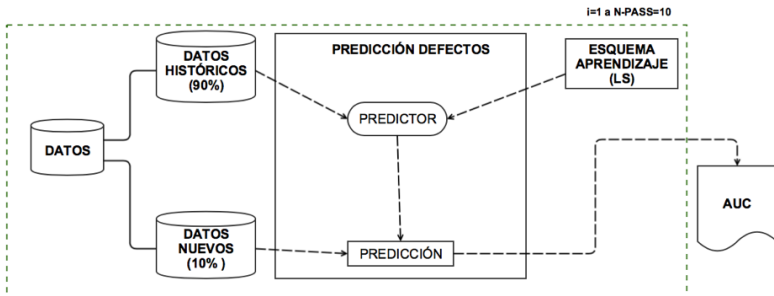


(Song et al., 2011)

Componente predictor (1/2)



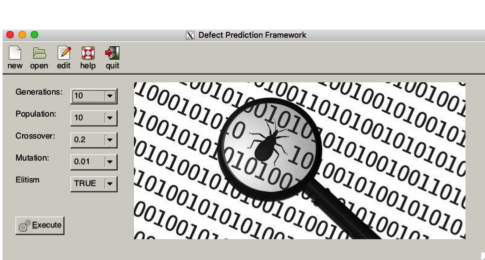
Componente predictor (2/2)



Caracterización: Revisión sistemática

- Se ejecutó una **revisión sistemática de literatura** siguiendo la **metodología de Kitchenham**. Los estudios analizados estuvieron entre el **2002-2017**. Se seleccionaron **120** estudios y de estos se analizaron **100**.
- Dentro de los autores más destacados están: Khofhgoftaar, Jang, Catal, **Malhotra**, **Menzies** y **Song** que representan el 60 % de los más frecuentes.
- La combinación de métricas con mayor frecuencia fueron **Halstead**, **McCabe** y **LOC** con 68,57 %.
- Las técnicas y modelos de minería más comunes fueron las de **aprendizaje máquina** con 47,14 %.

Diseño e implementación del marco de trabajo



Evaluación: genético vrs (Song & Menzies)

¿El desempeño de la aproximación genética propuesta fue similar a las de otros marcos de trabajos?

12 combinaciones - **Resultados similares a la literatura**

864 combinaciones - **Mejores resultados que la literatura**

Evaluación: genético vrs exhaustivo

¿El desempeño de la aproximación genética se comportó de forma similar a la de una aproximación exhaustiva?

864 combinaciones - **Resultados similares al exhaustivo**

¿El tiempo de ejecución de la aproximación genética fue similar a la de una aproximación exhaustiva?

864 combinaciones - **Mejores resultados que el exhaustivo**

Evaluación: configuraciones genéticas(1/3)

Se empleó un análisis de sensibilidad considerando los siguientes factores y sus niveles (Srinivas,2014):

- Población y generaciones (10, **20** y **40**)
- Mutación (**0.01**, 0.033 y 0.1)
- Cruce (**0.6**, 0.7 y 0.9)

Evaluación: Caso de la industria

- La aproximación genética presentó un desempeño mayor a un *random guessing* ($0.5 AUC$) con 7 módulos que reportaron al menos un fallo - AUC de 0.7586).
- La correlación entre la cantidad mínima de módulos que fallaron y el desempeño fue positiva para ambos escenarios.
- Experiencia de aplicar el marco de trabajo en la industria.

Período: 2015-2017

Cuadro: Publicaciones

Publicación	Tipo	BD
Software fault prediction: A Systematic Mapping Study	CibSE-Conferencia(2015)	Scopus-Elsevier
A software defect-proneness prediction framework: A new approach using genetic algorithms to generate learning schemes.	SEKE-Conferencia(2015)	CORE B
An Empirical Validation of an Automated Genetic Software Effort Prediction Framework using ISBSG Dataset. XIX Ibero-American Conference on Software Engineering	CibSE-Conferencia(2016)	Scopus-Elsevier
An Empirical Validation of Learning Schemes using an Automated Genetic Defect Prediction Framework	Iberamia-Conferencia(2016)	Springer-Verlag
An Empirical Validation of Learning Schemes using an Automated Defect Prediction Framework using Genetic Algorithms: A Validation of Empirical Studies	Iberamia-Journal(2016)	Scopus-Elsevier
An Empirical Evaluation of NASA-MDP Data sets using a Genetic Defect-Proneness Prediction Framework	CONCAPAN-Conferencia(2016)	IEEE-Xplore
A genetic algorithm based framework for software effort prediction	JSERD-Journal(2017)	Springer-OpenAccess

Conclusiones (1/2)

- Los **algoritmos genéticos** permiten construir **esquemas de aprendizaje** de forma **automática**, explorando un **espacio de búsqueda** mayor al de la literatura en un **tiempo razonable**.
- La **generación automática** de **esquemas de aprendizaje** requiere de considerable **experimentación** para lograr un alto desempeño.
- Entre más se **incrementa** el **espacio de búsqueda** hay más opción de encontrar una **mejor solución**.

Conclusiones (2/2)

- La **comparación** con una aproximación **exhaustiva** dio como resultados un **desempeño similar** y **mejores tiempos de respuesta**.
- El **estudio** de **sensibilidad** es importante para la obtención de **esquemas de aprendizaje** con mejor **desempeño**.
- El caso de estudio en Intel, sección CR, ratifica que es indispensable seguir probando el marco de trabajo en entornos privados.

Trabajo futuro

- Estudiar una **estrategía de automatización** de los **parámetros** de cada **técnica y algoritmo** utilizado en el esquema de aprendizaje.
- Incorporar más **técnicas y algoritmos** para los esquemas de aprendizaje. Es decir **ampliar el espacio de búsqueda**.
- Calcular la **densidad de defectos** en lugar de la **propensidad de defectos**.
- **Adaptar** el método para **predecir** en otros dominios, por ejemplo **estimación de esfuerzo**.

Agradecimientos:

