

# JCSA | 2021

12 y 13 de noviembre de 2021



LIBRO DE ACTAS



Libro de Actas de las Cuartas Jornadas de Calidad de Software y Agilidad / Gladys Noemí Dapozo, Emanuel Irrazábal, María de los Ángeles Ferraro, Horacio D. Kuna, Eduardo Zamudio, Alice Rambo, César Acuña, Verónica Bollati y Noelia Pinto ; compilación de Gladys Noemí Dapozo ; Emanuel Agustín Irrazábal. - 1a ed compendiada. - Corrientes : Universidad Nacional del Nordeste. Facultad de Ciencias Exactas, 2021. Libro digital, PDF

Editorial de la Universidad Nacional del Nordeste (EUDENE)

ISBN 978-987-3619-72-4

1. Software. 2. Jornadas. 3. Argentina. I. Dapozo, Gladys Noemí, comp. II. Irrazábal, Emanuel Agustín, comp.

CDD 004.0711

Fecha de catalogación: 03/01/2022

## **Autoridades**

### **Universidad Nacional del Nordeste**

Rectora: Prof. María Delfina Veiravé

Vicerrector: Dr. Mario H. Urbani

### **Facultad de Ciencias Exactas y Naturales y Agrimensura**

Decana: Mgter. María Viviana Godoy Guglielmone

Vicedecano: Dr. Enrique Laffont

### **Universidad Nacional de Misiones**

Rectora: Mgter. Alicia Violeta Bohren

Vicerrector: Ing. Fernando Luis Kramer

### **Facultad de Ciencias Exactas, Químicas y Naturales**

Decano: Dr. Luis Brumobsky

Vicedecano: Dr. Marcelo Marinelli

### **Universidad Tecnológica Nacional**

Rector: Ing. Héctor Aiassa

Vicerrector: Ing. Haroldo Avetta

### **Facultad Regional Resistencia**

Decano: Jorge A. De Pedro

Vicedecano: Dr. Ing. Walter Gustavo Morales

## **IV Jornadas de Calidad de Software y Agilidad**

Este evento es organizado en forma conjunta por las universidades de la región que tienen unidades académicas que ofrecen carreras de Informática, UNNE, UTN y UNaM, a través de un acuerdo específico institucional, con el objetivo de difundir avances significativos en el campo de conocimiento de la Ingeniería de Software, Calidad y Agilidad; y propiciar el encuentro entre las universidades, las empresas y los organismos del Estado para contribuir al desarrollo de la industria del software en la región de influencia de las universidades participantes. Las Jornadas de Calidad de Software y Agilidad (JCSA) se inician en el año 2017, bajo el nombre original de Jornadas de Calidad de Software, en esta edición, las jornadas se consolidaron como un foro regional de referencia, ampliando su espectro para enfatizar también temas relacionados al uso de la agilidad. Además, se propuso la publicación de los trabajos académicos presentados, previa evaluación de pares, que se incorporan en este libro de actas. Durante el evento se realizaron tres talleres vinculados con los temas de la Jornada, se expusieron los artículos y posters académicos aceptados y se presentaron experiencias de la industria del software, con la participación de empresas y organismos del Estado. La conferencia inaugural denominada “Estrategias pruebas de aceptación para Entrega Continua”, estuvo a cargo de Diego Fontdevilla (UNTREF). Las actividades propuestas estuvieron destinadas a ingenieros y licenciados en sistemas, estudiantes y docentes de estas especialidades, profesionales y empresarios del sector del software y servicios informáticos, así como también público interesado en la temática. En total participaron más de 200 personas en las actividades programadas, lo que evidencia el interés que estos temas suscitan en los destinatarios.

## **Comité Organizador**

Mgter. Gladys Noemí Dapozo  
(FaCENA - UNNE)

Dr. Emanuel Irrazábal  
(FaCENA - UNNE)

Lic. María de los Ángeles Ferraro  
(FaCENA - UNNE)

Dr. Horacio D. Kuna  
(FQCEyN - UNaM)

Dr. Eduardo Zamudio  
(FQCEyN - UNaM)

Ing. Alice Rambo  
(FQCEyN - UNaM)

Dr. César Acuña  
(FRRe - UTN)

Dra. Verónica Bollati  
(FRRe - UTN)

Dra. Noelia Pinto  
(FRRe - UTN)

## Comité de Programa

Mgter. Cristina Greiner (GICS - FaCENA - UNNE)	Dr. Diego Godoy (UGD)
Mgter. Gladys Noemí Dapozo (GICS - FaCENA - UNNE)	Ing. Edgardo Belloni (UGD)
Dr. Emanuel Irrazábal (GICS - FaCENA - UNNE)	Mag. Liliana Cuenca Pletsch (CINApTIC - UTN - FRRe)
Dr. Rubén Bernal (GICS - FaCENA - UNNE)	Dr. César J. Acuña (CINApTIC - UTN - FRRe)
Dr. David la Red Martínez (FaCENA - UNNE)	Dra. Verónica Bollati (CINApTIC - UTN - FRRe / CONICET)
Dra. Sonia Mariño (FaCENA - UNNE)	Dra. Noelia Pinto (CINApTIC - UTN - FRRe)
Mgter. M. Viviana Godoy Guglielmone (FaCENA - UNNE)	Esp. Gabriela Tomaselli (CINApTIC - UTN - FRRe)
Mgter. Mónica Tugnarelli (FCAD - UNER)	Ing. Nicolas Tortosa (CINApTIC - UTN - FRRe)
Dra. Gabriela Arévalo (DCyT - UNQ)	Ing. Valeria Sandobal Verón (GIESIN- UTN - FRRe)
Dra. María Fernanda Golobisky (UTN - FRStaFe)	Ing. Germán Gaona (CINApTIC - UTN - FRRe)
Master Ariel Pasini (LIDI - UNLP)	Dr. Horacio Leone (INGAR - UTN - FRSF)
Mgter. Pablo Thomas (LIDI - UNLP)	Dr. Silvio Gonnet (INGAR - UTN - FRSF)
Dr. Fernando Emmanuel Frati (DCByT - UNdeC)	Dr. Gustavo Rossi (LIFIA - UNLP)
Dra. Marcela Genero Bocco (Grupo Alarcos - UCLM)	Dra. Alejandra Garrido (LIFIA - UNLP)
Dr. Jorge Andrés Díaz Pace (ISISTAN - CONICET)	Dr. Andrés Rodríguez (LIFIA - UNLP)
Dr. Nazareno Aguirre (FCEQyN - UNRC / CONICET)	Dr. Marcelo Estayno (UNSAM)
Dra. Nancy Ganz (IIDII-FCEQyN - UNaM)	Dr. Luis Olsina (GIDIS - UNLPam)
Esp.Ing.Alice Rambo (IIDII-FCEQyN - UNaM)	Dra. Luciana Ballejos (CIDISI - UTN - FRSF)
Ing. Selva Nieves Ivaniszyn (FCEQyN - UNaM)	Dra. Luciana Roldan (INGAR - UTN - FRSF)
Lic. Sergio Caballero (FCEQyN - UNaM)	Dra. Milagros Gutierrez (UTN - FRSF)
Lic. Martín Rey (IIDII-FCEQyN - UNaM)	Dra. Mariel Alejandra Ale (CIDISI - UTN - FRSF)
Dr.Eduardo Zamudio (IIDII - FCEQyN - UNaM)	Dra. Elsa Estevez (UNSur / CONICET)
Dr.Horacio Kuna (IIDII - FCEQyN - UNaM)	Dra. Alicia Mon (UNLaM)

# Índice de Artículos

<b>Evaluación de conjuntos de datos utilizados en la construcción de modelos para la predicción de defectos en clases de proyectos software . . . . .</b>	<b>7</b>
Juan Carruthers, Celeste Ojeda	
<b>Evaluación de la calidad de visualizaciones de datos basada en el Proceso Analítico Jerárquico y Redes Neuronales de Propagación hacia Atrás . . . . .</b>	<b>17</b>
Andrea Lezcano, Joaquín Acevedo, Laura Godoy	
<b>Sonar JUploader, aplicación para el análisis, sincronización y actualización automática de proyectos a Sonar Cloud . . . . .</b>	<b>27</b>
Juan Pinto, Juan Carruthers, Emanuel Irrazabal	
<b>Storytelling Aplicado al Diseño de Sitios Web: Un Estudio de Caso . . . . .</b>	<b>37</b>
Andrea Lezcano, David Sandoval, Emanuel Irrazabal	
<b>Evaluación de la calidad en aplicaciones de realidad virtual: Una revisión sistemática de la literatura . . . . .</b>	<b>47</b>
Alice Raquel Rambo, Fabio Alberto Ledesma, Ruben Oscar Syniuk	
<b>Análisis y Gestión de Riesgo en Proyectos Software . . . . .</b>	<b>55</b>
Sergio Daniel Caballero, Horacio Daniel Kuna	
<b>i-QuAGI: Un enfoque inteligente para la gestión de calidad en proyectos de software ágiles . . . . .</b>	<b>64</b>
Nicolás Tortosa, Jazmin Teng, Lucas Maidana, Cesar Acuña, Noelia Pinto	
<b>Emociones en el uso de software: Una experiencia de relevamiento utilizando PrEmo . . . . .</b>	<b>74</b>
Gabriela Tomaselli, Nicolas Alegre, Rodrigo Cuevas, Cesar Acuña, Noelia Pinto	
<b>Desarrollo de entornos educativos interactivos digitales mediante Design Thinking . . . . .</b>	<b>84</b>
Alejandro Micheloud, Eduardo Zamudio	
<b>A relationship between ISO 9001:2015 and agility practices . . . . .</b>	<b>94</b>
Alvaro Ruiz de Mendarozqueta, Fabio Bustos, Pedro E. Colla	
<b>Análisis del uso de frameworks en la estimación de la duración de proyectos web . . . . .</b>	<b>105</b>
Yanina Medina, Gladys Dapozo	





## Evaluación de conjuntos de datos utilizados en la construcción de modelos para la predicción de defectos en clases de proyectos software

Juan Andrés Carruthers, Celeste Ojeda Rodríguez

Grupo de Investigación en Calidad de Software, Facultad de Ciencias Exactas y Naturales y  
Agrimensura, Universidad Nacional del Nordeste  
jacarruthers@exa.unne.edu.ar  
malu221098@gmail.com

**Resumen.** La predicción de defectos consiste principalmente en la identificación de componentes de software con mayor probabilidad de errores para una asignación de recursos efectiva. Esto se realiza por medio de modelos predictivos que son entrenados con datos de los proyectos, como información del proceso de desarrollo o el producto software. En este estudio, se presentan tres clasificadores entrenados con tres conjuntos de datos conformados por valores de métricas orientadas a objetos obtenidas del análisis estático del código de cinco sistemas Java. Para el entrenamiento de cada modelo se utiliza el método de ensamble validación y votación configurado con los algoritmos Regresión Logística, Función de Base Radial, Árbol de Decisión, Perceptrón Multicapa, Naïve Bayes y Máquinas de Vector Soporte. Los modelos predictivos registran resultados favorables, con valores mayores a 0.8 de exactitud y mayores a 0.87 de especificidad.

**Palabras Clave:** Predicción de defectos, Métricas orientadas a objetos, Mantenimiento del software

### 1 Introducción

El mantenimiento del software dedica una importante cantidad de recursos dentro del proceso de desarrollo, siendo en algunos casos una de las etapas que más presupuesto requiere [1]. La reparación de defectos es una actividad central en el mantenimiento del software, pero es necesario identificarlos previamente. De esta manera surgen los modelos de predicción de defectos, donde el escenario principal es la asignación de recursos a los componentes de software que requieren ser revisados [2]. El tiempo y la mano de obra son recursos finitos, por lo que tiene sentido asignar personal y recursos a áreas de un sistema de software con una mayor cantidad probable de errores.

La predicción de defectos ha generado un interés generalizado durante un período de tiempo considerable y numerosos artículos han reportado diferentes enfoques implementados. Desde la utilización de métricas de cambio en la historia del proyecto en [2], la información de defectos previos en [3] o el contexto del proceso de desarrollo en [4].

Otro enfoque para la predicción de defectos asume que el diseño y el comportamiento actuales del programa influyen en la presencia de defectos futuros. Estos enfoques no requieren la historia del sistema, sino que analizan su estado actual con más detalle, utilizando una variedad de métricas como el caso de [3], [5] y [6]. De esta manera, se busca responder a la siguiente pregunta de investigación:

**RQ:** ¿Qué conjunto de métricas orientadas a objetos obtenidas de una versión del sistema tienen mejor rendimiento para predecir defectos en las clases de un proyecto software?

Para ello se proponen tres modelos clasificadores con el último enfoque mencionado. Se emplean métricas obtenidas del análisis estático del código fuente de Chidamber y Kemerer (CK) [7] y un conjunto de métricas orientadas a objetos (OO) para entrenar los clasificadores e información de defectos para la creación de las etiquetas. Un clasificador es entrenado solamente con las métricas de CK, el segundo con las métricas OO y el tercero con una combinación de ambos conjuntos de datos (CK + OO). La intención es determinar los predictores más efectivos para la detección de clases propensas a tener defectos.

El resto del artículo se encuentra organizado de la siguiente manera. En la sección 2, están los trabajos relacionados. En la sección 3 y 4 se presentan la descripción detallada del diseño del experimento y los resultados obtenidos respectivamente. Finalmente, en la sección 5 se encuentran las discusiones y conclusiones del trabajo desarrollado.

## 2 Trabajos relacionados

La predicción de defectos en artefactos software ha sido estudiado por una gran cantidad de autores a lo largo del tiempo. En la literatura referente se pueden encontrar diferentes enfoques y fuentes de datos empleados para estos fines. Por ejemplo, Zimmermann et al. [3] proponen un modelo de regresión logística con la información de defectos, métricas de complejidad y cantidad de estructuras específicas halladas en el código de tres versiones de lanzamiento de Eclipse a nivel de archivos y paquetes.

D'Ambros y Robbes [2] evalúan seis enfoques en cinco sistemas de fuente libre, para ello reunieron medidas de cambios de la historia del proyecto; la entropía de estos cambios; información de defectos previos; métricas del código fuente de una versión; múltiples versiones; y la entropía entre estas versiones. A diferencia del trabajo de Zimmermann et al., D'Ambros y Robbes analizan los sistemas a nivel de clases en vez de paquetes dado que estas son unidades más pequeñas y por ende su revisión requiere menor trabajo. Además, en el caso que fuera estrictamente necesario generar los valores a nivel de paquete es posible recrearlos desde las clases que integran cada paquete. Jureczko y Madeyski [8] también usan valores de 19 métricas de múltiples versiones, reportadas como indicadores de buena calidad en otros estudios.

Por su parte Zhang et al. [4] consideran el contexto de desarrollo del sistema para la predicción de defectos teniendo en cuenta factores tales como el lenguaje de programación, el uso de un sistema de rastreo de defectos, cantidad totales de líneas de código,

de archivos, de commits y de desarrolladores. Ghotra et al. [5] entrenan modelos clasificadores con el conjunto de datos NASA [9] utilizando distintos algoritmos tales como Naïve Bayes, Árboles de Decisión, Función de Base Radial, Máquinas de Vector Soporte, entre otros. El conjunto de datos NASA contiene valores de 40 métricas de código fuente de una versión.

Antecedentes más actuales, como el de Pecorelli y Nucci [6], también emplean el conjunto de datos publicado por Jureczko y Madeyski [8] para evaluar el desempeño de clasificadores entrenados con seis técnicas de ensamble: bagging, boosting, validación y votación, selección adaptativa de clasificadores, COmbined DEfect Predictor (CODEP) [10] y bosque aleatorio. Qiao et al. [11] implementan y evalúan un enfoque de aprendizaje profundo para la predicción de defectos con un conjunto de datos compuesto por métricas de complejidad y tamaño del código fuente.

### **3 Experimento**

A continuación, se describen el conjunto de datos, los métodos, las técnicas, las herramientas y las métricas de desempeño para desarrollar y evaluar el experimento propuesto.

#### **3.1 Conjunto de datos**

El conjunto de datos utilizado fue obtenido de [12], publicado inicialmente en [2]. Es un repositorio público usado con fines científicos en varios estudios [13 - 15]. El mismo contiene información evolutiva, de una versión y de reportes de defectos de las clases de cinco sistemas Java de fuente abierta. Eclipse JDT Core es el conjunto de herramientas núcleo del entorno de desarrollo integrado Eclipse para la construcción de software. Eclipse PDE UI, también es un conjunto de herramientas de desarrollo, pero para la creación de extensiones de Eclipse. Equinox Framework es un marco de trabajo que implementa el estándar OSGi. Lucene es una biblioteca específica para la indexación y búsqueda de cadenas de texto. Mylyn es un marco de trabajo para la gestión del ciclo de vida del sistema para Eclipse.

Se emplearon los valores de métricas estructurales de CK, OO y cantidad de defectos de una sola versión a nivel de clases de los sistemas anteriormente mencionados para entrenar los modelos. Los defectos provienen de los registros de los sistemas de rastreo de defectos Bugzilla y Jira de los sistemas anteriormente mencionados. La Tabla 1 muestra con un mayor nivel de detalle estas métricas.

**Tabla 1.** Descripción de las métricas. (Fuente [2]).

Tipo	Sigla de la métrica en inglés	Descripción
CK	CBO	Acoplamiento entre objetos
	DIT	Profundidad de árbol de herencia
	LCOM	Falta de cohesión en métodos
	NOC	Cantidad de hijos
	RFC	Respuesta por clase
	WMC	Peso de métodos por clase
OO	FanIn	Cantidad de clases diferentes que hacen referencia a la clase
	FanOut	Cantidad de clases diferentes referenciadas por la clase
	NOA	Cantidad de atributos
	NOAI	Cantidad de atributos heredados
	LOC	Cantidad de líneas de código
	NOM	Cantidad de métodos
	NOMI	Cantidad de métodos heredados
	NOPRA	Cantidad de atributos privados
	NOPRM	Cantidad de métodos privados
	NOPA	Cantidad de atributos públicos
	NOPM	Cantidad de métodos públicos

El conjunto de datos fue dividido en dos partes. Por un lado, las métricas de CK y por otro las demás métricas OO. Con ellos se entrenaron tres clasificadores. El primero utilizó los datos de CK, el segundo los demás datos de métricas OO y el tercero la combinación de estos dos (CK + OO). En base a la cantidad de defectos por clase se crearon las etiquetas con los valores 0 y 1 para los clasificadores binarios. Si el número de defectos en una clase era mayor a 0 se asignó un 1, y en el caso contrario un 0.

### 3.2 Datos desbalanceados

Generalmente la predicción de defectos es un problema desbalanceado, porque normalmente el número de clases defectuosas en un sistema es mucho menor que las no defectuosas. Este conjunto de datos no es la excepción, siendo que las clases defectuosas representa menos del 16% del total. Dado que esto puede sesgar el rendimiento del clasificador del modelo predictor [16] se aplicó la técnica de muestreo Synthetic Minority Over-sampling TEchnique (SMOTE) [17] para asegurar que los conjuntos de datos de entrenamiento tengan una proporción similar de clases defectuosas que no defectuosas. Se utilizó la implementación de SMOTE de la biblioteca de Python imbalanced-learn [18].

### 3.3 Algoritmos y herramientas

Siguiendo las recomendaciones de [6], se entrenaron los modelos con el algoritmo de ensamble validación y votación. Validación y votación [19] (también denominada votación) combina las puntuaciones de confianza obtenidas por los clasificadores subyacentes. Para cada instancia a predecir, cada clasificador devuelve una puntuación de confianza que varía entre 0 y 1. Un operador combina las puntuaciones (por ejemplo, promedio o máximo). Una clase se marca con defectos si la combinación de las puntuaciones de confianza es superior a 0.5, mientras que, en caso contrario, se predice como sin defectos. El método de votación fue configurado usando los siguientes clasificadores: Regresión Logística, Función de Base Radial, Árbol de Decisión, Perceptrón Multicapa, Naïve Bayes y Máquinas de Vector Soporte.

Tanto el método de votación como los demás clasificadores fueron implementados con la biblioteca de Python Scikit-learn [20].

### 3.4 Métricas de evaluación

Para evaluar el rendimiento de los modelos de predicción, se calcularon las matrices de confusión como se muestra en la Tabla 2. En la matriz de confusión, el verdadero positivo (VP) es el número de clases defectuosas que se predicen correctamente como defectuosas; falso negativo (FN) cuenta el número de clases defectuosas que se predice incorrectamente como no defectuosas; falso positivo (FP) mide la cantidad de clases que están libre de defectos, pero se predice incorrectamente como defectuosas; y verdadero negativo (VN) representa el número de clases sin defectos que se predicen correctamente como no defectuosas.

**Tabla 2.** Matriz de confusión.

<b>Real \ Predicho</b>	<b>Defectuosa</b>	<b>No defectuosa</b>
Defectuosa	Verdadero positivo (VP)	Falso negativo (FN)
No defectuosa	Falso positivo (FP)	Verdadero negativo (VN)

Se calcularon las métricas precisión, exhaustividad y exactitud al igual que [3] y [8], para evaluar el desempeño de los clasificadores. También se incluyeron los valores obtenidos de la métrica de especificidad como en [21] y coeficiente de correlación de Matthew como en [6].

**Precisión.** Relaciona el número de verdaderos positivos (predichos y observados como propensos a defectos) con el número de clases predichas como propensas a defectos.

$$\text{Precisión} = \frac{VP}{VP+FP} \quad (1)$$

Un valor cercano a uno es deseable y significaría que todas las clases que se predijeron que tenían defectos en realidad tenían defectos.

**Exhaustividad.** Relaciona el número de verdaderos positivos (predichos y observados como propensos a defectos) con el número de clases que realmente tenían defectos.

$$Exhaustividad = \frac{VP}{VP+FN} \quad (2)$$

Un valor cercano a uno es el mejor y significaría que se predijo que todas las clases que tenían defectos observados tenían defectos.

**Especificidad.** Relaciona el número de verdaderos negativos (predichos y observados como no propensos a defectos) con el número de clases que no tenían defectos.

$$Especificidad = \frac{VN}{VN+FP} \quad (3)$$

Un valor cercano a uno es el mejor y significaría que se predijo que todas las clases que no tenían defectos observados efectivamente no los tenían.

**Exactitud.** Relaciona el número de clasificaciones correctas (verdaderos positivos y verdaderos negativos) con el número total de clases.

$$Exactitud = \frac{VP+VN}{VP+VN+FP+FN} \quad (4)$$

Un valor de uno es lo mejor y significaría que el modelo se clasificó perfectamente, es decir, no cometió un solo error.

**Coefficiente de correlación de Matthew (CCM).** Indica hasta qué punto las variables independientes y dependientes están relacionadas entre sí.

$$CCM = \frac{VN \times VP - FP \times FN}{\sqrt{(VN+FN)(FP+VP)(VN+FP)(FN+VP)}} \quad (5)$$

Los valores van de -1 a 1, donde los valores cercanos a 1 indican un mayor rendimiento.

## 4 Resultados

Tal y como se mencionó en la sección 3, se entrenaron tres clasificadores con conjuntos de datos diferentes CK, OO y CK + OO. En cada caso se aplicó la técnica de sobre-muestreo SMOTE para la generación de nuevos casos en los datos de entrenamiento para evitar sesgar los resultados. Los modelos fueron construidos con el método de votación con seis algoritmos clasificadores. Finalmente se computaron las métricas exactitud, precisión, exhaustividad, especificidad y CCM para evaluar el rendimiento de los modelos.

En la Tabla 3 se encuentran los resultados obtenidos de las métricas de desempeño de los tres clasificadores. En cada caso se puede observar una exactitud mayor a 0.8 y

ninguno exhibe una diferencia significativa; por otra parte CCM presenta índices similares para cada clasificador, siendo CK y CK + OO los que tienen valores más altos con 0.331 y 0.324 respectivamente.

Con respecto a la precisión, el clasificador CK + OO presenta el mejor desempeño de los tres con 0.508. Continuando con exhaustividad, OO registró el mejor valor con 0.476, diferenciándose en más de 0.14 comparado con CK. Finalmente, la especificidad no presenta valores dispares entre los clasificadores, siendo CK + OO el que obtuvo mayor puntaje con 0.93.

**Tabla 3.** Resultados de métricas de desempeño de los clasificadores.

Clasificador	Precisión	Exhaustividad	Especificidad	Exactitud	CCM
CK	0.455	0.329	0.927	0.834	0.331
OO	0.404	0.476	0.87	0.808	0.294
CK + OO	0.508	0.35	0.933	0.837	0.324

## 5 Amenazas a la validez

En esta sección, discutimos las amenazas que podrían afectar la validez del estudio empírico realizado en este artículo.

### 5.1 Validez de constructo

Las amenazas en esta categoría se refieren a la relación entre la teoría y la observación, es decir, las variables medidas pueden no medir realmente la variable conceptual. La amenaza es el ruido que afecta a los repositorios de Bugzilla. En [22] Antoniol et al. mostró que una fracción considerable de los informes de problemas marcados como errores en Bugzilla son de hecho "no errores", es decir, problemas no relacionados con el mantenimiento correctivo. De igual manera los autores del conjunto de datos inspeccionaron y verificaron manualmente una muestra estadísticamente significativa (107) de los errores de Eclipse JDT Core que más del 97% de ellos eran errores reales. Por tanto, el impacto de esta amenaza en los experimentos es limitado.

### 5.2 Validez de la conclusión

Está vinculado con la relación entre el tratamiento y el resultado. Para asegurar que los resultados no hubieran sido sesgados por efectos de confusión debidos al desequilibrio de datos se adoptó el procedimiento SMOTE [17].

En cuanto a la evaluación del desempeño de los modelos experimentados, se consideró CCM, el cual ha sido muy recomendado [23, 24] para interpretar correctamente los resultados.

### 5.3 Validez externa

Relacionada con la generalización de los hallazgos. Se han aplicado técnicas de predicción solo a sistemas de software de código abierto y ciertamente existen diferencias entre el desarrollo industrial y el de código abierto. Los autores minimizaron esta amenaza mediante el uso de partes de Eclipse, un sistema que, si bien es de código abierto, tiene una sólida formación industrial. Una segunda amenaza se refiere al lenguaje: todos los sistemas de software están escritos en Java. Incluir sistemas que no sean Java aumentaría su valor, pero introduciría problemas ya que los sistemas necesitarían ser procesados por diferentes analizadores, produciendo resultados variables.

Para disminuir el impacto de una tecnología o herramienta específica, los autores del conjunto de datos incluyeron sistemas desarrollados con diferentes sistemas de control de versiones y diferentes sistemas de rastreo de defectos (Bugzilla y Jira). Además, los sistemas de software del conjunto de datos son desarrollados por equipos de desarrollo independientes y surgieron del contexto de dos comunidades no relacionadas (Eclipse y Apache).

## 6 Discusiones y conclusiones

La asignación de recursos humanos y económicos a los componentes de software defectuosos es una tarea esencial para el mantenimiento del sistema. Por lo tanto, también resulta esencial definir cuáles son los atributos que predicen donde están estos componentes. Este es objetivo principal de este artículo, por medio de la siguiente pregunta de investigación: ¿qué conjunto de métricas orientadas a objetos obtenidas de una versión del sistema tienen mejor rendimiento para predecir defectos en las clases de un proyecto software? Para ello, se construyeron y evaluaron tres modelos predictivos, utilizando en cada caso un conjunto de datos diferente.

En la Tabla 3 del presente trabajo, los tres modelos tienen una exactitud mayor a 0.8. Comparando con experiencias previas como [3], donde el máximo valor registrado fue de 0.789, todos los modelos hicieron un mejor trabajo de clasificación. Pero, ciertamente esta métrica no es suficiente para describir en profundidad el desempeño de los clasificadores, como se puede observar en los valores de precisión y exhaustividad en 5 de los 6 casos se registra un valor menor a 0.5.

La precisión y exhaustividad representan qué tan bien se identificaron las clases defectuosas. Dicho esto, ningún clasificador se destacó significativamente con respecto al resto en el proceso de identificación de clases con defectos. Sin embargo, los tres modelos han detectado exitosamente las clases sin defectos, esto es confirmado tanto por los valores de exactitud y especificidad en todos los modelos, con resultados mayores a 0.8. Esto no es atribuible a la sobre proporción de clases sin defectos de las muestras originales, porque, como se mencionó en la sección 3, se aplicó el método de muestreo SMOTE para evitar este sesgo.

Finalmente, se puede mencionar que en la mayoría de las métricas de desempeño CK + OO ha obtenido mejores resultados exceptuando CCM y exhaustividad. Por lo tanto, se puede afirmar que utilizar solamente las métricas OO para predecir defectos



podría implicar en un peor desempeño del clasificador, y sería preferible emplearla complementariamente con CK, o inclusive reemplazarla por completo.

## Referencias

1. Erlikh, L. Leveraging legacy system dollars for e-business. *IT Prof.* 2, 17–23 (2000).
2. D'Ambros, M., Lanza, M. & Robbes, R. An extensive comparison of bug prediction approaches. in *Proceedings - International Conference on Software Engineering* 31–41 (2010). doi:10.1109/MSR.2010.5463279.
3. Zimmermann, T., Premraj, R. & Zeller, A. Predicting defects for eclipse. in *Proceedings - ICSE 2007 Workshops: Third International Workshop on Predictor Models in Software Engineering, PROMISE'07* 9–9 (IEEE, 2007). doi:10.1109/PROMISE.2007.10.
4. Zhang, F., Mockus, A., Keivanloo, I. & Zou, Y. Towards building a universal defect prediction model with rank transformed predictors. *Empir. Softw. Eng.* 21, 2107–2145 (2016).
5. Ghotra, B., McIntosh, S. & Hassan, A. E. Revisiting the impact of classification techniques on the performance of defect prediction models. in *Proceedings - International Conference on Software Engineering* vol. 1 789–800 (IEEE Computer Society, 2015).
6. Pecorelli, F. & Di Nucci, D. Adaptive selection of classifiers for bug prediction: A large-scale empirical analysis of its performances and a benchmark study. *Sci. Comput. Program.* 205, 102611 (2021).
7. Chidamber, S. R. & Kemerer, C. F. Towards a metrics suite for object oriented design. *ACM SIGPLAN Not.* 26, 197–211 (1991).
8. Jureczko, M. & Madeyski, L. Towards identifying software project clusters with regard to defect prediction. in *ACM International Conference Proceeding Series 1* (ACM Press, 2010). doi:10.1145/1868328.1868342.
9. Shepperd, M., Song, Q., Sun, Z. & Mair, C. Data quality: Some comments on the NASA software defect datasets. *IEEE Trans. Softw. Eng.* 39, 1208–1215 (2013).
10. Panichella, A., Oliveto, R. & De Lucia, A. Cross-project defect prediction models: L'Union fait la force. in *2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering, CSMR-WCRE 2014 - Proceedings* 164–173 (IEEE Computer Society, 2014). doi:10.1109/CSMR-WCRE.2014.6747166.
11. Qiao, L., Li, X., Umer, Q. & Guo, P. Deep learning based software defect prediction. *Neurocomputing* 385, 100–110 (2020).
12. Bug prediction dataset, <https://bug.inf.usi.ch>, último acceso: 16/09/2021
13. D'Ambros, M., Lanza, M. & Robbes, R. Evaluating defect prediction approaches: A benchmark and an extensive comparison. in *Empirical Software Engineering* vol. 17 531–577 (Springer, 2012).
14. Jing, X., Wu, F., Dong, X., Qi, F. & Xu, B. Heterogeneous cross-company defect prediction by unified metric representation and CCA-based transfer learning. in *2015 10th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, ESEC/FSE 2015 - Proceedings* 496–507 (Association for Computing Machinery, Inc, 2015). doi:10.1145/2786805.2786813.
15. Xu, Z., Liu, J., Yang, Z., An, G. & Jia, X. The Impact of Feature Selection on Defect Prediction Performance: An Empirical Comparison. in *Proceedings - International Symposium on Software Reliability Engineering, ISSRE* 309–320 (IEEE Computer Society, 2016). doi:10.1109/ISSRE.2016.13.

16. Bennin, K. E., Keung, J., Phannachitta, P., Monden, A. & Mensah, S. MAHAKIL: Diversity Based Oversampling Approach to Alleviate the Class Imbalance Issue in Software Defect Prediction. *IEEE Trans. Softw. Eng.* 44, 534–550 (2018).
17. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* 16, 321–357 (2011).
18. Imbalanced-learn Documentation, <https://imbalanced-learn.org/stable/>, último acceso 16/09/2021.
19. Kittler, J., Hatef, M., Duin, R. P. W. & Matas, J. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* 20, 226–239 (1998).
20. Scikit-learn, <https://scikit-learn.org/stable/>, último acceso 16/09/2021.
21. Rathore, S. S. & Kumar, S. Software fault prediction based on the dynamic selection of learning technique: findings from the eclipse project study. *Appl. Intell.* (2021) doi:10.1007/s10489-021-02346-x.
22. Antoniol, G., Ayari, K., Di Penta, M., Khomh, F. & Guéhéneuc, Y. G. Is it a bug or an enhancement? A text-based approach to classify change requests. *Proc. 2008 Conf. Cent. Adv. Stud. CASCON'08* (2008) doi:10.1145/1463788.1463819.
23. Hall, T., Beecham, S., Bowes, D., Gray, D. & Counsell, S. Developing fault-prediction models: What the research can show industry. *IEEE Softw.* 28, 96–99 (2011).
24. Yao, J. & Shepperd, M. Assessing software defection prediction performance: Why using the Matthews correlation coefficient matters. *ACM Int. Conf. Proceeding Ser.* 120–129 (2020) doi:10.1145/3383219.3383232.

## Evaluación de la calidad de visualizaciones de datos basada en el Proceso Analítico Jerárquico y Redes Neuronales de Propagación hacia Atrás

Andrea Lezcano Airaldi<sup>1</sup>, Joaquín Acevedo<sup>1</sup>, María Laura Godoy<sup>1</sup>

<sup>1</sup> Universidad Nacional del Nordeste, Corrientes, Argentina  
{alezcano, jacevedo}@exa.unne.edu.ar, marialgodoy15@gmail.com

**Resumen.** La visualización de datos se ha vuelto esencial para comprender grandes conjuntos de datos y comunicar hallazgos. La narrativa basada en datos se utiliza cada vez más para dar soporte a la toma de decisiones en diferentes ámbitos, por lo cual resulta fundamental evaluar su calidad. Este trabajo propone la implementación del proceso analítico jerárquico en combinación con redes neuronales de propagación hacia atrás (AHP-BP), teniendo en cuenta el conocimiento y experiencia de los evaluadores. Se analizaron estudios existentes para conocer los factores de mayor influencia en la calidad de visualizaciones narrativas y se elaboró el sistema de índices de evaluación. Se seleccionaron diferentes usuarios como expertos evaluadores y se calculó el peso de cada uno. Se ajustaron los valores iniciales de las evaluaciones con los pesos de los expertos y luego se ponderaron para mitigar la subjetividad de los cálculos. Finalmente, con los resultados de la evaluación de 20 visualizaciones se entrenó un modelo de red neuronal BP. Los resultados demuestran que el método refleja la competencia de los expertos y reduce la subjetividad de la evaluación. La capacidad de aprendizaje de la red neuronal proporciona resultados precisos y confiables.

**Palabras clave:** proceso analítico jerárquico, redes neuronales, visualización de datos, evaluación de calidad.

### 1 Introducción

La visualización de datos se ha vuelto esencial para comprender grandes conjuntos de datos y comunicar hallazgos. La narrativa basada en datos o «data storytelling» se utiliza cada vez más para la toma de decisiones[1], por lo cual es importante considerar las perspectivas de los usuarios finales en la construcción de dichas narrativas.

En los últimos años, la evaluación se convirtió en un eje central en el campo de las visualizaciones de datos. Hay diversos métodos cualitativos y cuantitativos para evaluar diferentes aspectos de las visualizaciones narrativas[2]. Algunos de estos incluyen experimentos controlados y pruebas de usabilidad[3].

El proceso analítico jerárquico [4] es un método para toma de decisiones que permite estructurar un problema como una jerarquía de criterios con distintos “pesos” y seleccionar los más adecuados a través de comparaciones por pares realizadas por expertos,

lo que permite traducir opiniones subjetivas en métricas precisas. El método tradicional de AHP evalúa únicamente a través de las valoraciones de los expertos. Sin embargo, cuando estos tienen diferentes conocimientos, competencias y experiencias, la valoración de cada uno frente a un mismo criterio puede no ser igual y la equidad del cálculo se pierde. Por ello, es importante considerar estas diferencias e incorporarlas a los cálculos de ponderación.

La red neuronal de propagación hacia atrás (*back propagation*, BP) es un modelo típico de red neuronal de retroalimentación multicapa compuesto por una capa de entrada, una capa oculta y una capa de salida. Fue propuesto por Rumelhart y McClelland en 1986 y es uno de los algoritmos de red neuronal más utilizados[5][6].

El objetivo de este trabajo es implementar un modelo de evaluación que combine el método analítico jerárquico con redes neuronales de propagación hacia atrás, asegurando un análisis efectivo de los factores que influyen en la calidad de las visualizaciones y reduciendo la subjetividad de los expertos.

Se analizó la literatura existente para conocer los factores de mayor influencia en la calidad de visualizaciones narrativas y se elaboró el sistema de índices de evaluación. Se seleccionaron diferentes usuarios como expertos evaluadores y se calculó el peso de cada uno. Se ajustaron los valores iniciales de las evaluaciones con los pesos de los expertos y luego se ponderaron para mitigar la subjetividad de los cálculos. Finalmente, con los resultados de la evaluación de 20 visualizaciones se entrenó un modelo de red neuronal de propagación hacia atrás.

El trabajo se encuentra organizado de la siguiente manera. En la Sección 2 se describen los antecedentes y los trabajos relacionados. En la Sección 3 se enumeran las decisiones tomadas para la metodología de investigación propuesta, la recolección y el análisis de los datos, mientras que en la Sección 4 se detallan los resultados y en la Sección 5 se presenta las conclusiones.

## **2 Antecedentes y Trabajos Relacionados**

### **2.1 Evaluación de calidad de visualizaciones de datos**

Las visualizaciones de datos efectivas se elaboran en base al mensaje que los autores pretenden comunicar y considerando el mejor medio para mostrar las variables [7][8]. Actualmente existen varios enfoques para evaluar la efectividad de las visualizaciones de datos. Los métodos tradicionales incluyen estudios de percepción gráfica [9][10], experimentos controlados, o estudios de usabilidad [11].

Lam et. al. [12] definen diferentes tipos de escenarios de evaluación, categorizados en aquellos para comprender los procesos de análisis de datos y aquellos que evalúan las visualizaciones en sí mismas. Basan su categorización en preguntas y objetivos, animando así a considerar el contexto antes de elegir un método de evaluación.

Tory y Moller [13] proponen que la opinión de los expertos en el dominio podría ser un complemento útil para los experimentos controlados, tanto para la evaluación heurística de la usabilidad como para comprender las actividades cognitivas de alto nivel. Sin embargo, no solo las opiniones de los expertos en el dominio, sino también las

opiniones de los especialistas visuales, como artistas, diseñadores gráficos e ilustradores, pueden ser beneficiosas, como se demuestra en [14].

## 2.2 El Proceso Analítico Jerárquico

El Proceso Analítico Jerárquico (AHP) fue propuesto por Saaty [4] en 1980 y permite determinar los pesos de criterios estructurados jerárquicamente. Los diferentes criterios se comparan por pares y los resultados de las comparaciones se organizan en una matriz. La comparación es cualitativa e indica si un criterio es más significativo que el otro y a qué nivel pertenece la prioridad. La técnica utilizada permite convertir las estimaciones cualitativas obtenidas de los expertos en valores cuantitativos.

Dado que la matriz de evaluación se construye mediante puntajes de expertos, para verificar el sesgo de subjetividad, se colocan los pesos relativos en la fórmula de relación de consistencia (RC) para la prueba de consistencia. Cuando esta es menor que 0,1, la matriz de evaluación es aceptable. El AHP es útil cuando hay muchos intereses involucrados y varias personas participan en el proceso de evaluación [4].

## 2.3 Redes neuronales de propagación hacia atrás

La red neuronal BP es un modelo de red neuronal de retroalimentación multicapa compuesto por una capa de entrada, una capa oculta y una capa de salida. La estructura del modelo y los pesos de la red neuronal BP se obtienen a través del proceso de aprendizaje. El mismo se divide en dos etapas: una etapa de feed forward que calcula la entrada y salida reales de cada nodo y la etapa de retro propagación, donde el peso de cada conexión se corrige a lo largo de la ruta inversa para reducir el error [15]. La idea básica de la red neuronal BP es lograr el error cuadrático mínimo de la capa de salida, ajustando los pesos y umbrales de la red, de modo que el valor de salida sea lo más cercano posible al valor objetivo. La estructura de la red se muestra en la Fig. 1.

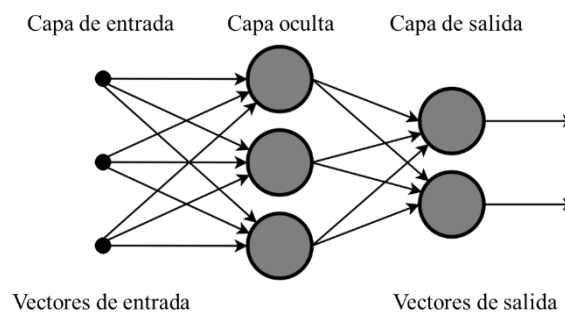


Fig. 1. Diagrama de la estructura básica de una red neuronal BP.

## 2.4 Combinación de AHP y redes neuronales BP

Dado que el principal inconveniente del AHP es la evaluación subjetiva, muchos autores han intentado mejorarlo combinándolo con otros métodos como la lógica difusa

[16] o máquinas de vector soporte [17]. En cuanto a las redes neuronales BP, si los datos brutos se utilizan directamente para entrenar el modelo, con el aumento de datos, la estructura de la red neuronal BP se vuelve más compleja y el tiempo de entrenamiento es más largo, lo que resulta en una reducción del rendimiento [18].

Atendiendo a estos problemas, Li et. al. [19] proponen el método AHP mejorado, en el cual se tienen en cuenta aspectos de la personalidad de los expertos para corregir los pesos de las evaluaciones una primera vez y luego se combinan los resultados con una red neuronal BP. En [20] se considera la perspectiva o intereses de los expertos.

Esta corrección puede evitar la disminución del rendimiento de la red neuronal BP debido a la redundancia de datos, y la combinación de métodos puede corregir los errores causados por la subjetividad de los expertos.

### **3 Metodología**

Los pasos seguidos para la evaluación de la calidad de las visualizaciones fueron los siguientes [5][15][19][20]:

1. Construcción del sistema de índices de evaluación de la calidad de visualizaciones;
2. Cálculo de los pesos relativos de los índices en todos los niveles mediante AHP. Se utilizó un sistema en línea [21] para garantizar la ejecución sin errores.
3. Cálculo del peso de la evaluación de cada experto;
4. Ponderación y promedio global de los valores de los índices junto con las evaluaciones de los expertos de los pasos 2 y 3;
5. Entrenamiento del modelo de red neuronal de BP tomando el resultado del paso anterior como entrada.
6. Verificación y análisis de la aplicación.

#### **3.1 Selección de índices de evaluación**

Se identificaron los criterios más importantes que influyen en la calidad de las visualizaciones de datos a partir de los resultados de estudios relacionados con narrativa basada en datos y buenas prácticas de visualización [22]-[30]. Los mismos se clasificaron en cinco categorías: Narrativa, Diseño, Interacción, No Manipulación y Gráficos Apropriados. La Narrativa hace referencia a cómo se cuenta la historia e incluye el orden de la secuencia de eventos (N1) y la incorporación de los elementos básicos de una historia (N2). El Diseño, se compone de prácticas relativas a la visualización propiamente dicha: usar colores consistentes (D1), resaltar lo importante (D2), eliminar el desorden visual (D3), y usar texto, etiquetas y anotaciones para facilitar la comprensión (D4). La categoría Interacción incluye el incentivo a la exploración (I1), y la estimulación de la curiosidad del usuario (I2). La No Manipulación se refiere a la ética al crear visualizaciones. En este sentido, abarca las prácticas de ser fiel a la información (M1), no citar información fuera de contexto (M2), variar los datos, no el diseño (M3) y no distorsionar los gráficos (M4). Finalmente, la categoría Gráficos Apropriados apunta a la elección de un gráfico simple (G1) y a eliminar la complejidad innecesaria (G2). Las categorías y subcategorías constituyen los índices de primer y segundo nivel

respectivamente. El sistema de índices resultante se muestra en la Tabla 1, e incluye cinco índices de primer nivel y 14 índices de segundo nivel.

**Tabla 1.** Índices de evaluación de la calidad de visualizaciones de datos.

Objetivo	Primer Nivel	Segundo Nivel
Calidad	Narrativa	N1 - Ordenar la secuencia de eventos
		N2 - Incorporar elementos básicos de la historia
		D1 - Usar colores consistentes
	Diseño	D2 - Resaltar lo importante
		D3 - Eliminar el desorden visual
		D4 - Usar texto, etiquetas y anotaciones
	Interacción	I1 - Incentivar la exploración
		I2 - Estimular la curiosidad del usuario
		M1 - Ser fiel a la información
	No Manipulación	M2 - No citar información fuera de contexto
		M3 - Variar los datos, no el diseño
		M4 - No distorsionar los gráficos
	Gráfico Apropiado	G1 - Elegir un gráfico simple
		G2 - Eliminar complejidad innecesaria

### 3.2 Cálculo del peso en cada nivel utilizando AHP

Se invitó a 10 usuarios (expertos) a calificar los índices de evaluación, y se comparó la puntuación de los expertos en pares. Se calculó la matriz de evaluación de cada experto en cada nivel. Tomando como ejemplo a uno de los expertos, la matriz de evaluación de los cinco criterios de primer nivel se muestra en la Tabla 2. Se calculó la razón de consistencia (RC) de la matriz de evaluación. El valor fue de 0.079, por lo que la matriz tiene una consistencia aceptable. Este procedimiento se repitió con cada experto.

**Tabla 2.** Matriz de evaluación

	Narrativa	Diseño	Interacción	No Manipulación	Gráfico Apropiado
Narrativa	1	1	1/2	1/3	1
Diseño	1	1	3	1/2	1
Interacción	2	1/3	1	1/2	1
No Manipulación	3	2	2	1	1
Gráfico Apropiado	1	1	1	1	1

### 3.3 Cálculo del peso de cada experto

Los expertos fueron divididos en tres grupos, de acuerdo con sus conocimientos, competencias y experiencias con visualizaciones de datos: usuarios avanzados, usuarios intermedios y usuarios ocasionales. De acuerdo con su categoría, se le asignó a cada experto un peso de evaluación y luego se ponderaron y promediaron las estimaciones realizadas en el paso anterior utilizando sus pesos de evaluación.

### 3.4 Peso ajustado de los índices

El peso global en todos los niveles del sistema de índices se obtiene mediante el método de ponderación y promedio de los valores obtenidos mediante AHP junto con el vector de ponderación de los expertos. Los resultados se muestran en la Tabla 3.

**Tabla 3.** Peso global de los criterios de calidad para visualizaciones.

Primer nivel	Segundo Nivel	Tercer nivel	Peso
Calidad	Narrativa	N1 - Ordenar la secuencia de eventos	0,102
		N2 - Incorporar elementos básicos de la historia	0,059
		D1 - Usar colores consistentes	0,035
	Diseño	D2 - Resaltar lo importante	0,056
		D3 - Eliminar el desorden visual	0,068
		D4 - Usar texto, etiquetas y anotaciones	0,053
		I1 - Incentivar la exploración	0,083
	Interacción	I2 - Estimular la curiosidad del usuario	0,083
		M1 - Ser fiel a la información	0,046
	No Manipulación	M2 - No citar información fuera de contexto	0,06
		M3 - Variar los datos, no el diseño	0,071
		M4 - No distorsionar los gráficos	0,092
	Gráfico Apropiado	G1 - Elegir un gráfico simple	0,096
		G2 - Eliminar complejidad innecesaria	0,096

### 3.5 Red neuronal BP

Para determinar el peso de los factores de calidad los valores obtenidos por el AHP mejorado se utilizan como muestras para entrenar y probar la red neuronal BP, de modo



que los pesos fijos de cada factor se pueden determinar una vez que la red alcanza estabilidad. Los pasos específicos, siguiendo las definiciones dadas por [20] son:

- Inicializar los pesos y umbrales de la red. El vector peso de la capa de entrada a la capa oculta es  $V_k = (v_{qj,k})_{Q \times J}$ , en el que  $v_{qj,k}$  representa el peso desde la  $j$ -ésima capa de entrada al  $q$ -ésimo nodo oculto y  $\theta_{qk}$  es el umbral del  $q$ -ésimo nodo de la capa oculta. El vector peso de la capa de entrada a la capa implícita es  $W_k = (w_{mq,k})_{M \times Q}$ , en el que  $w_{mq,k}$  representa el peso desde el  $q$ -ésimo nodo de la capa oculta al  $m$ -ésimo nodo de salida y  $\theta_{mk}$  es el umbral del  $m$ -ésimo nodo de la capa de salida.
- Ingresar la muestra de aprendizaje y calcular los valores de salida de cada capa. El vector de valores esperados de la capa de salida  $T_i = (t_{i1}, t_{i2}, \dots, t_{iM})$  es generado en base a los resultados de evaluación de los expertos, cuando el vector de datos normalizados  $X_i$  de la  $i$ -ésima muestra es ingresado. El vector salida de la capa de salida es  $H_{i,k} = (h_{i1,k}, h_{i2,k}, \dots, h_{iQ,k})$  y el vector de salida de la capa de salida es  $O_{i,k} = (o_{i1,k}, o_{i2,k}, \dots, o_{iM,k})$ .

## 4 Resultados

### 4.1 Preprocesamiento de los datos

Se seleccionaron veinte visualizaciones de datos cuya calidad fue evaluada por los diez expertos de acuerdo con el sistema de índices de evaluación propuesto en la Sección 3.1. El vector de evaluación es  $X$ , el vector de ponderación de evaluación es  $W$  y el vector de resultado de evaluación  $Z$  de las 20 visualizaciones se calcula mediante la fórmula  $Z = WX$ .

$Z = [0.4120, 0.6950, 0.7188, 0.3601, 0.4237, 0.7112, 0.1720, 0.3239, 0.9107, 0.2914, 0.3527, 0.1251, 0.6548, 0.2101, 0.6511, 0.1056, 0.6119, 0.8621, 0.5651, 0.4402]$

### 4.2 Modelado de la red neuronal BP.

Para el diseño, entrenamiento y pruebas de la red neuronal BP se utilizó la librería de Scikit-learn [31] para redes neuronales de Python. El número de nodos de la capa de entrada de la red se estableció en  $n=14$ , el número de nodos de salida fue  $M=14$  de acuerdo con el vector de valores esperados y el número de nodos de la capa oculta se estableció en  $j = 10$ . La tasa de aprendizaje se estableció en 0.05, con un paso de entrenamiento de 20.000 veces y la precisión de convergencia del error de entrenamiento en  $10^{-2}$ .

Seleccionando los primeros 10 elementos del vector de evaluación como el conjunto de datos de entrenamiento, la variable de entrada es  $X_t = [X_1, X_2, \dots, X_{10}]^T$ . Los valores de la evaluación de los criterios de calidad ( $Z_T = [Z_1, Z_2, \dots, Z_{10}]^T$ ) son usados como valor de salida esperado de la red, y el vector de salida esperado es  $t_T = [t_1, t_2, \dots, t_{10}]^T$ .

Seleccionando los últimos 10 elementos del vector de evaluación como el conjunto de datos de entrenamiento, la variable de entrada es  $X_v = [X_{11}, X_{12}, \dots, X_{20}]^T$ . Los

valores de la evaluación de los criterios de calidad ( $Z_v = [Z_{11}, Z_{12}, \dots, Z_{20}]^T$ ) son usados como valor de salida esperado de la red, y el vector de salida esperado es  $t_v = [t_{11}, t_{52}, \dots, t_{20}]^T$ .

### 4.3 Entrenamiento de la red neuronal BP

Después de 50 pasos de cálculo, la varianza máxima de los datos de simulación es 0.0098, el error cuadrático medio de entrenamiento de la red es menor que el valor objetivo  $10^{-2}$ . Este resultado satisface los requisitos de precisión en el aprendizaje. Es decir, los parámetros seleccionados por el modelo cumplen los requisitos. El error relativo del resultado del entrenamiento se muestra en la Tabla 4. Se puede observar que el valor de salida de la red está muy cerca del valor esperado.

**Tabla 4.** Resultados del entrenamiento de la red neuronal BP

	1	2	3	4	5	6	7	8	9	10
Valor esperado	0.4120	0.6950	0.7188	0.3601	0.4237	0.7112	0.1720	0.3239	0.9107	0.2914
Valor de entrenamiento	0.4152	0.7041	0.7190	0.3700	0.4245	0.7190	0.1790	0.3255	0.9019	0.2944
Error relativo	-0,0032	-0,0091	-0,0002	-0,0099	-0,0008	-0,0078	-0,0070	-0,0016	0,0088	0,0030

### 4.4 Prueba de la red neuronal BP

El resultado de la evaluación de los datos de la muestra 11-20 se comparó con el valor de salida esperado. El error se muestra en la Tabla 5. Se puede observar que el error entre el valor de salida y el valor esperado es pequeño.

Los resultados de la evaluación de la red neuronal BP muestran que el modelo tiene una alta eficiencia y un error pequeño y puede ser utilizado en la evaluación de la calidad visualizaciones.

**Tabla 5.** Resultados de la prueba de la red neuronal BP

	51	52	53	54	55	56	57	58	59	60
Valor esperado	0.3501	0.1299	0.6448	0.2050	0.6547	0.1047	0.6203	0.8580	0.5766	0.4393
Valor de entrenamiento	0.3527	0.1251	0.6548	0.2101	0.6511	0.1056	0.6119	0.8621	0.5651	0.4402
Error relativo	-0,0026	0,0048	-0,01	-0,0051	0,0036	-0,0009	0,0084	-0,0041	0,0115	-0,0009

## 5 Conclusiones

En este trabajo, se propone la aplicación del método AHP mejorado para la evaluación de la calidad de visualizaciones de datos. Se analizaron una serie de estudios existentes referentes a narrativa basada en datos y buenas prácticas de visualización y en base a los resultados del análisis se confeccionó el sistema de índices de evaluación. Se

seleccionaron diferentes grupos de usuarios como expertos evaluadores y se calculó el peso de cada experto en función de sus competencias, experiencia y conocimiento respecto a visualizaciones. Se ajustaron los valores iniciales de los índices con los pesos de los expertos y luego se ponderaron para mitigar la subjetividad de los cálculos. Finalmente, con los resultados de la evaluación de 20 visualizaciones se entrenó un modelo de red neuronal BP.

Se comprobó que el algoritmo AHP mejorado puede reducir la subjetividad de los expertos en la evaluación, y la combinación con una red neuronal BP permite realizar una evaluación más precisa y efectiva.

## Referencias

- [1] W. Willett, J. Heer, J. M. Hellerstein, and M. Agrawala, "CommentSpace: Structured support for collaborative visual analysis," in *Conference on Human Factors in Computing Systems - Proceedings*, 2011, pp. 3131–3140, doi: 10.1145/1978942.1979407.
- [2] S. Carpendale, "Evaluating information visualizations," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4950 LNCS, pp. 19–45, 2008, doi: 10.1007/978-3-540-70956-5\_2.
- [3] C. Plaisant, "The challenge of information visualization evaluation," *Proc. Work. Adv. Vis. Interfaces AVI*, pp. 109–116, 2004, doi: 10.1145/989863.989880.
- [4] T. L. Saaty, "The Analytic Hierarchy Process: Planning, priority setting, resource allocation", McGraw-Hill, 1980. Doi: [https://doi.org/10.1016/0377-2217\(82\)90022-4](https://doi.org/10.1016/0377-2217(82)90022-4).
- [5] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning internal representation by error propagation. *Neurocomputing: Foundations of research*. Cambridge: MIT Press. 14.
- [6] Bailer-Jones, C. A. L., Gupta, R., Singh, H. P., & Bulsari, A. B. (2001). An introduction to artificial neural networks. *Neural Networks for Chemical Engineers*, 19(12), 36–62.
- [7] Few S. Effectively Communicating Numbers: Selecting the Best Means and Manner of Display [White Paper]. Secondary Effectively Communicating Numbers: Selecting the Best means and manner of Display [White Paper]; 2005. [http://www.perceptualedge.com/articles/Whitepapers/Communicating\\_Numbers.pdf](http://www.perceptualedge.com/articles/Whitepapers/Communicating_Numbers.pdf)
- [8] Cleveland W, McGill R. Graphical perception: theory, experimentation, and application to the development of graphical methods. *J Am Stat Assoc* 1984; 79 (387): 531–54.
- [9] J. Heer and M. Bostock. Crowdsourcing graphical perception: using Mechanical Turk to assess visualization design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 203–212, 2010.
- [10] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualization. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 1303–1312, 2009.
- [11] S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, volume 4950, pages 19–45. Springer LNCS, 2008.
- [12] H. Lam *et al.*, "Empirical Studies in Information Visualization: Seven Scenarios To cite this version: Empirical Studies in Information Visualization: Seven Scenarios," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 9, pp. 1520–1536, 2012.

- [13] M. Tory and T. Moller. Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, Sept./Oct. 2005. doi>10.1109/MCG.2005.102
- [14] D. Acevedo, C. D. Jackson, F. Drury, and D. H. Laidlaw. Using visual design experts in critique-based evaluation of 2D vector visualization methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):877–884, July/Aug. 2008. doi>10.1109/TVCG.2008.29
- [15] Fu, J., & Zhang, L. (2012). Green marketing performance evaluation of circular economy based on AHP and BP neural network model. *Science and Technology Management Research*, 20, 222–227.
- [16] Akbar, M. A., Shameem, M., Khan, A. A., Nadeem, M., Alsanad, A., & Gumaei, A. (2020). A fuzzy analytical hierarchy process to prioritize the success factors of requirement change management in global software development. *Journal of Software: Evolution and Process*. doi:10.1002/smr.2292
- [17] Wei, Wang. (2011). Network security assessment of AHP and SVM combination. *Computer Simulation*, 28(3), 182–185
- [18] Wang, X., Liang, D., Song, W., & Zhou, Y. (2018). Distributed intelligent pension system based on bp neural network. *Wireless Personal Communications*, 1–10, 1–12
- [19] Li, W., Xu, G., Xing, Q. et al. Application of Improved AHP-BP Neural Network in CSR Performance Evaluation Model. *Wireless Pers Commun* 111, 2215–2230 (2020). <https://doi.org/10.1007/s11277-019-06981-z>
- [20] Liu, Y. (2018) ‘An improved AHP and BP neural network method for service quality evaluation of city bus’, *Int. J. Computer Applications in Technology*, Vol.58,No.1, p.37–44
- [21] Goepel, K.D. (2018). Implementation of an Online Software Tool for the Analytic Hierarchy Process (AHP-OS). *International Journal of the Analytic Hierarchy Process*, Vol. 10 Issue 3 2018, pp 469-487, <https://doi.org/10.13033/ijahp.v10i3.590>
- [22] C. Nussbaumer Knafloc, *Storytelling with Data: A data visualization guide for business professionals*. Hoboken, New Jersey: John Wiley & Sons, Ltd., 2015.
- [23] R. Kosara, J. Mackinlay, and T. Software, “Storytelling: The Next Step for Visualization.”
- [24] R. Kosara, “An Argument Structure for Data Stories.”
- [25] E. Segel and J. Heer, “Narrative Visualization: Telling Stories with Data,” 2010.
- [26] C. Tong et al., “Storytelling and visualization: A survey,” *VISIGRAPP 2018 - Proc. 13th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 3, pp. 212–224, 2018, doi: 10.5220/0006601102120224.
- [27] C. Tong et al., “Storytelling and visualization: An extended survey,” *Inf.*, vol. 9, no. 3, Mar. 2018, doi: 10.3390/info9030065.
- [28] J. Boy, F. Detienne, and J. D. Fekete, “Storytelling in information visualizations: Does it engage users to explore data?,” *Conf. Hum. Factors Comput. Syst. - Proc.*, vol. 2015-April, pp. 1449–1458, 2015, doi: 10.1145/2702123.2702452.
- [29] C. Nussbaumer Knaflic, “Data Stories.”
- [30] E. R. Tufte, *The Visual Display of Quantitative Information*.
- [31] Scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation <https://scikit-learn.org/stable/index.html>. Accessed: 2021-07-16

## **Sonar JUploader, aplicación para el análisis, sincronización y actualización automática de proyectos a Sonar Cloud**

Juan Alberto Pinto Oppido, Juan Andrés Carruthers, Emanuel Agustín Irrazábal

Grupo de Investigación en Calidad de Software, Facultad de Ciencias Exactas y Naturales y  
Agrimensura, Universidad Nacional del Nordeste  
juan\_al\_pinto@hotmail.com  
{jacarruthers, eirrazabal}@exa.unne.edu.ar

**Resumen.** Sonar Cloud es una plataforma que permite el monitoreo del código fuente verificando el cumplimiento de reglas, detección de defectos en el código, definición del nivel de cobertura, entre otras. Cada una de estas características en el código fuente son cuantificadas o medidas por medio de un conjunto métricas, y Sonar Cloud ofrece una herramienta llamada Sonar Scanner que puede integrarse a otros desarrollos para implementar funcionalidades de análisis estático de código con la plataforma. Sin embargo, esta herramienta no está diseñada para analizar un lote de proyectos automáticamente, siendo esto un problema al momento de generar métricas para estudios empíricos del software. De esta manera surge la aplicación Sonar Juploader que busca brindar estas facilidades a los grupos de investigación. En el siguiente artículo se detalla Sonar Juploader, una aplicación java que permite analizar automáticamente un lote de proyectos con Sonar Cloud. La aplicación da soporte a la gestión y uso de organizaciones de Sonar Cloud, pre-configuración de proyectos, análisis de proyectos, y visualización de reportes.

**Palabras Clave:** Sonar Cloud, Análisis lotes de proyectos, Herramienta, Análisis estático código fuente.

### **1 Introducción**

En la Ingeniería de Software se trabaja mayoritariamente con la "construcción de aplicaciones software multi-versión" [1]. Por lo tanto, muchas de las actividades asociadas con una aplicación software provocan revisiones para mejorar la funcionalidad o para corregir errores, especialmente en las metodologías ágiles [2]. En el caso del desarrollo software, la calidad puede estudiarse desde el punto de vista de: i) la calidad del proceso de desarrollo software, y ii) la calidad del código fuente [3]. En este último caso es necesario obtener métodos empíricos para demostrar la calidad del software [4] y utilizar evidencia directamente relacionada con el producto software resultante a partir de métricas e indicadores que se vinculen directamente con la calidad [5].

El uso masivo de repositorios para el código fuente (por ej., SourceForge, GitHub o Maven) le ha otorgado a los investigadores e ingenieros de software el acceso a millones de proyectos y, por lo tanto, datos para el desarrollo de estudios empíricos [6 - 8]. En este contexto, una práctica para demostrar la efectividad de las métricas como predictores de las características de calidad del software es la construcción de las denominadas colecciones de proyectos [9]. Estas colecciones son un insumo para los grupos de trabajo y sirven como mecanismo de comparación para distintos tipos de experimentos.

Sin embargo, dada la gran cantidad de proyectos que pueden estar contenidos en estas colecciones, la generación de métricas no resulta una tarea trivial. Aunque existan herramientas que facilitan el análisis automático del código fuente, este solamente es aplicado a un proyecto a la vez. Por ello es necesaria una herramienta que de soporte al proceso de análisis de una colección masiva de proyectos software.

Sonar JUploader es una aplicación java, cuya función principal es analizar lotes de proyectos de forma automática con Sonar Cloud, evitando la tarea manual al usuario de realizar numerosos análisis individuales, asistiendo en la pre-configuración de los mismos y obteniendo reportes y gráficos estadísticos en base a los análisis realizados.

El resto del artículo se encuentra organizado de la siguiente manera. En la sección 2, están los trabajos relacionados. En la sección 3 y 4 se presenta la aplicación desarrollada y su evaluación en un lote de proyectos respectivamente. Finalmente, en la sección 5 se encuentran las conclusiones del trabajo desarrollado.

## **2 Trabajos relacionados**

En los últimos años un gran número de herramientas para el análisis estático del código fuente han sido reportadas en estudios empíricos para la recolección de metadatos de proyectos software [10 – 12]. Estos metadatos en forma de métricas o indicadores revelan diferentes características del código fuente desde su tamaño, complejidad, acoplamiento, cohesión, entre otras [13]; proporcionando información acerca de la calidad del software en cuestión.

Algunos ejemplos de herramientas bastante recurrentes en la literatura para la generación de metadatos son CKJM [14] que calcula 8 métricas orientadas a objetos con el código compilado en Java. PMD [15] que evalúa la calidad del código por medio de distintas reglas reportando fallas en el código y métricas de diseño. Findbugs (ahora SpotBugs) [16] utiliza análisis estático para buscar bugs en el código Java. Sonar Qube o Cloud en sus versiones local y remota respectivamente [17], es una solución basada en la nube para análisis continuo de la calidad, fiabilidad y seguridad del código, detectando defectos, vulnerabilidades y errores.

Sin embargo, estas herramientas en su mayoría, carecen de funcionalidades que permitan su ejecución de manera iterativa a un lote de proyectos, siendo esto un limitante al momento de generar metadatos necesarios para realizar estudios empíricos del software. De esta manera surge la aplicación Sonar Juploader que busca brindar estas facilidades a los grupos de investigación.

### 3 Sonar Juploader

Sonar Juploader es una aplicación de escritorio construida con el lenguaje de programación java, que emplea la plataforma de análisis de Sonar Cloud a través de una herramienta a nivel de comandos, llamada Sonar Scanner [18]. Sonar Scanner puede ser utilizada para realizar el análisis del código fuente de un proyecto tras configurar y especificar manualmente una serie de parámetros del mismo.

#### 3.1 Sonar Scanner

Sonar Cloud brinda una serie de herramientas para realizar escaneo y análisis de proyectos en base al sistema de construcción utilizado en el mismo (Sonar Scanner for Gradle, Sonar Scanner for Maven, Sonar Scanner for Ant) o la plataforma/herramienta de automatización y gestión usada (Sonar Scanner for .Net, Sonar Scanner for Jenkins, Sonar Scanner for Azure Devops). Para realizar un análisis genérico que no se apoye en ninguna de estas tecnologías específicas, existe Sonar Scanner, el cual se adaptará al tipo de proyecto según el lenguaje de programación presente en los ficheros de código fuente.

Los requisitos mínimos para realizar un análisis con Sonar Cloud son los siguientes:

- Una cuenta en <https://sonarcloud.io/> donde crear una organización para subir los análisis realizados, y generar tokens de seguridad usados para obtener acceso a la organización desde fuera por la aplicación.
- Un proyecto software a analizar, previamente configurado con su respectivo fichero “sonar-project.properties” el cual contiene la clave del proyecto dentro de la organización, el nombre del proyecto, su versión, su lenguaje, la ubicación de sus ficheros de código fuente, la ubicación de sus ficheros compilados y la ubicación de sus librerías, entre otros datos.
- Contar localmente con la herramienta Sonar Scanner obtenida desde la web oficial desde <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

#### 3.2 Administración de organizaciones

Con la administración de organizaciones de Sonar Cloud el usuario es capaz de cargarlas, editarlas y borrarlas, y serán usadas a la hora de realizar análisis para evitar la carga reiterada de la misma información en cada oportunidad. En la Fig. 1 puede observarse la interfaz de administración de organizaciones, donde cada una posee un título identificatorio para facilitar su selección, una descripción para comprender su finalidad, el nombre clave dentro de la plataforma Sonar Cloud, su respectivo token de seguridad para el acceso, y la carpeta desde la cual se obtienen los proyectos a analizar, en caso de que esta no varíe, de lo contrario puede omitirse este dato.



**Fig. 1.** Interfaz de creación de organizaciones de Sonar JUploader.

### 3.3 Pre-configuración de proyectos

La pre-configuración de proyectos de Sonar Cloud requiere la creación del fichero “sonar-project.properties”, ubicado en la raíz del directorio del proyecto, previo a la realización de un análisis. En la Fig. 2 podemos observar la interfaz de pre-configuración donde podremos elegir una de las organizaciones creadas, y automáticamente se obtendrá la descripción y se generará el primer segmento del ProjectKey que la referencia. Esto se omite si se ingresa como invitado a la aplicación, debiendo editar la ProjectKey manualmente al no contar con organizaciones cargadas.

Luego, tras elegir la dirección donde se encuentra el proyecto a pre-configurar, el nombre de dicho proyecto se actualizará automáticamente en el ProjectKey y Project-Name, pero podemos editarlos si deseamos ajustarlos a otra estructura.



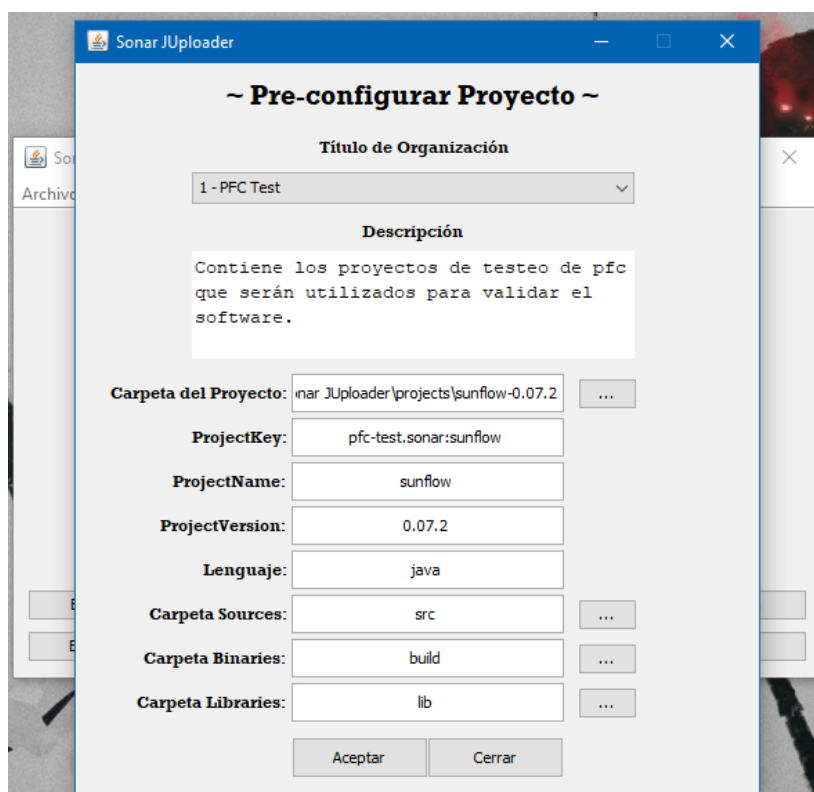


Fig. 2. Interfaz de pre-configuración de proyectos de Sonar JUploader.

Se introduce la versión del proyecto, el lenguaje de programación del mismo, y las ubicaciones de los ficheros de código fuente (sources, en java son ficheros .java), los ficheros compilados (binaries, en java son ficheros .class) y las librerías utilizadas por el mismo (libraries, en java son ficheros .jar).

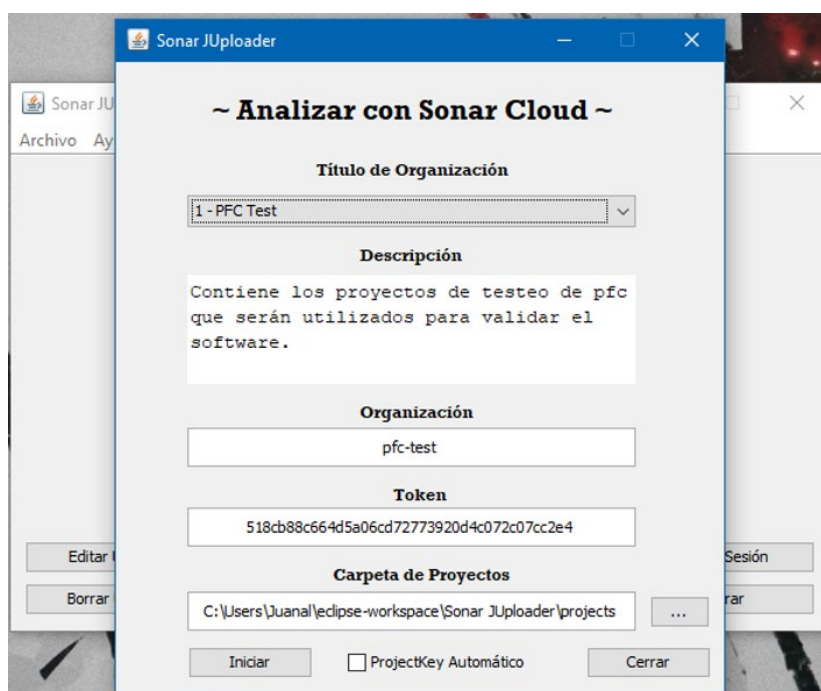
De forma alternativa, en caso de no querer especificar dichos directorios, predeterminadamente se carga automáticamente el carácter “.”, de manera que la herramienta Sonar Scanner tratará la carpeta del proyecto como base para todos los ficheros.

Una vez cargada toda la información requerida, al presionar el botón “Aceptar”, se generará el fichero de configuración “sonar-project.properties” respectivo, y dicho proyecto estará preparado para ser analizado.

### 3.4 Análisis de proyectos

La funcionalidad principal de la aplicación es la ejecución de análisis de proyectos, donde se automatizó el proceso de llamado del análisis; evitando la necesidad de usar la consola de comandos y permitiendo analizar cualquier número de proyectos sin más intervención, en lugar de sólo uno.

Como se visualiza en la Fig. 3, se puede elegir entre las organizaciones cargadas del usuario, y obtener automáticamente los datos para realizar el análisis. Esto se omite si se ingresa como invitado a la aplicación, debiendo ser suministrados manualmente al no contar con organizaciones cargadas.



**Fig. 3.** Interfaz de análisis de proyectos de Sonar JUploader.

Aun así, todos los campos son editables, en caso de requerirse la alteración de los valores para testeos, o análisis en diferentes directorios, en caso de que estos varíen de forma regular.

Al presionar el botón “Iniciar”, comenzará el análisis de todos los proyectos dentro del directorio, uno detrás de otro, como se mostró anteriormente, siendo luego subidos a la plataforma de Sonar Cloud donde podrán observarse los resultados. Puede marcarse la opción de “ProjectKey Automático” para casos en los que se analicen proyectos con una organización diferente a la que figura en el fichero de configuración sonar.properties, para evitar el trabajo de editarlos individualmente.

### 3.5 Visualización de reportes

Es posible consultar información acerca de los análisis ejecutados, por ejemplo, qué organización, fecha y hora, cuantos proyectos analizó, cuáles y la dirección de la carpeta en la cual se ubicaban, como puede observarse en la Fig. 4.

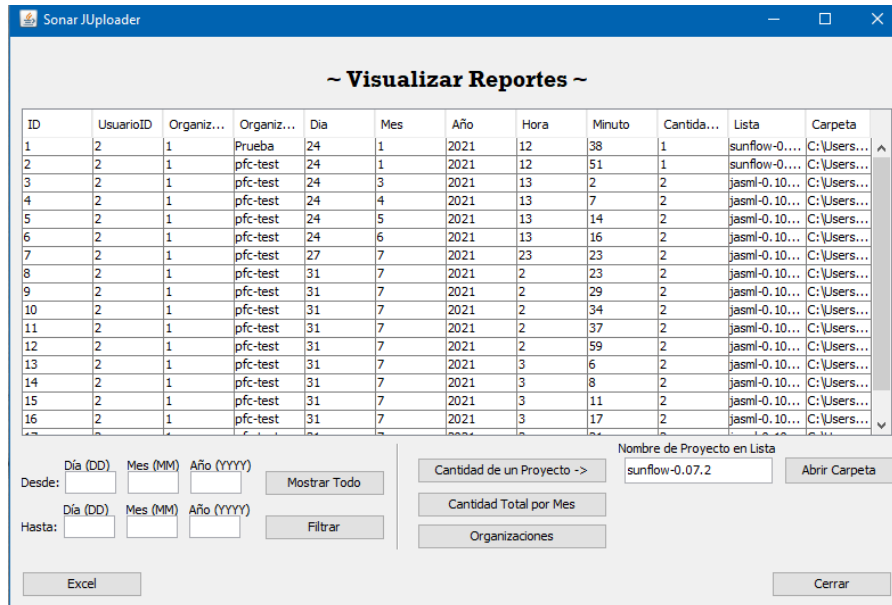


Fig. 4. Interfaz de visualización de reportes de Sonar JUploader.

Además, pueden filtrarse los datos para mostrar en la tabla los análisis realizados entre dos fechas especificadas, los registros mostrados pueden ser exportados a un fichero de Excel, y si se selecciona una fila correspondiente a un análisis, puede usarse el botón “Abrir Carpeta” para automáticamente dirigirnos al directorio utilizando el explorador de archivos.

El botón “Cantidad de un Proyecto” despliega un gráfico de barras con los análisis ejecutados de un proyecto específico mensualmente, y la cantidad total de análisis realizados al mismo como se visualiza en la Fig. 5.

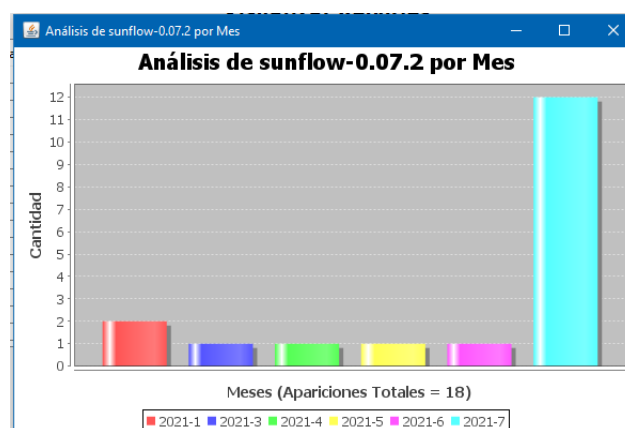


Fig. 5. Reporte de cantidad de un proyecto de Sonar JUploader.

De manera similar, los botones “Cantidad Total por Mes” y “Organizaciones” despliegan gráficos de barras del total de análisis ejecutados de un proyecto y total realizados por organización mensualmente.

## 4 Validación

Para validar la herramienta se crearon dos organizaciones en Sonar Cloud, Corpus-2021 [19] y PFC-Test [20], donde se registraron los resultados del análisis de una colección de proyectos software basada en el Qualitas Corpus [9]. Corpus-2021 [21] es la versión actualizada del Qualitas Corpus y contiene las versiones estables más recientes de cada proyecto dentro de la colección original. Esta colección de 110 proyectos codificados en java fue analizada, primero con Sonar Scanner y segundo utilizando Sonar Juploader; y los resultados fueron guardados en Corpus-2021 y PFC-Test respectivamente. Una vez finalizados los procesos de análisis, se calcularon los tiempos registrados en Sonar Cloud y fueron comparados para determinar la efectividad de la herramienta construida.

En la Fig. 6 se puede observar el listado de estos proyectos dentro de la organización de Sonar Cloud “Corpus-2021” [19], donde se encuentran algunos resultados de métricas evaluadas.

El primer análisis del lote de proyectos utilizando Sonar Scanner inició el 26 de mayo del 2021 a las 10:27 PM, y finalizó el 1 de junio del 2021 a las 4:16 AM; tomando así un total de 125 horas y 49 minutos, es decir, 69.26 minutos por proyecto. Es necesario aclarar que se descartó el tiempo del primer proyecto (Ant) porque el mismo fue analizado con mucha anterioridad (12 de marzo del 2021), y su inclusión hubiese distorsionado el total de tiempo transcurrido.

El segundo utilizando la herramienta Sonar JUploader, inició el 24 de agosto del 2021 a las 4:00 AM, y finalizó ese mismo día a las 12:28 PM; tomando así un total de 8 horas y 28 minutos. El resultado obtenido indicó que se habían analizado correctamente 106 de los 110 proyectos, habiendo incurrido en fallo solamente 4 de ellos, debido a la pérdida momentánea de la conexión de internet requerida para la sincronización con el servidor de Sonar Cloud. Se realizó luego un segundo análisis con los 4 proyectos faltantes, iniciando también el 24 de agosto del 2021 a las 8:21 PM, y finalizando ese mismo día a las 9:23 PM; totalizando una hora y 2 minutos. Sumando estos dos análisis, los 110 proyectos fueron analizados en 9 horas y 30 minutos, es decir, 5.18 minutos por proyecto.

Como se puede evidenciar se pudo reducir el tiempo promedio de análisis por proyecto de 69.26 minutos a 5.18 empleando Sonar Juploader, presentando una reducción de 64.08 minutos por proyecto. Además cabe remarcar que, en este último, el proceso no fue supervisado hasta su finalización, en cambio el primer análisis requirió un trabajo manual ininterrumpido. Se determina así que la herramienta es capaz de agilizar los tiempos necesarios para generar métricas de la plataforma Sonar Cloud de una gran cantidad de proyectos de forma automatizada.

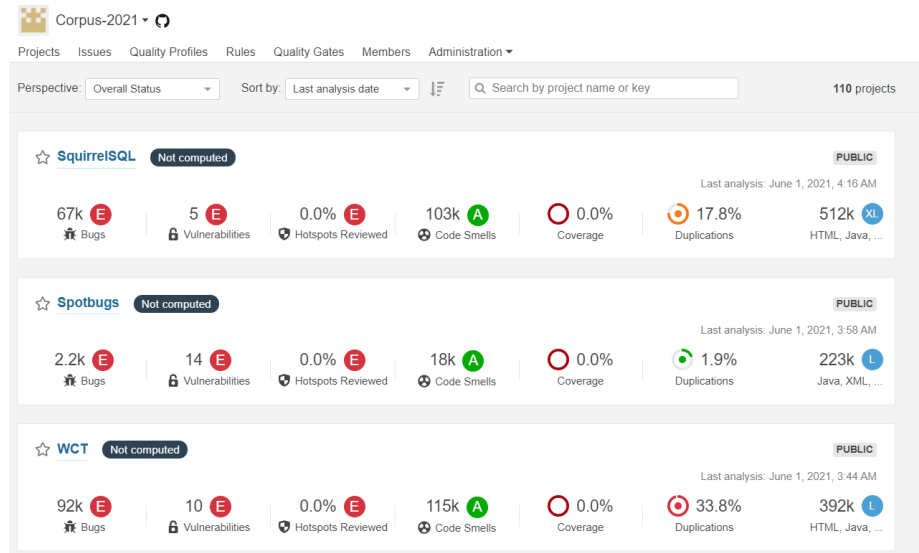


Fig. 6. Proyectos analizados visualizados en la organización corpus-2021.

## 5 Conclusión y futuros trabajos

Este artículo presenta Sonar Juploader, herramienta para el soporte en el cómputo de métricas de código fuente dirigida a los grupos de investigación en la calidad de software. Su funcionalidad principal consiste en el análisis de un lote de proyectos de Sonar Cloud de manera automatizada, y además permite administrar organizaciones de Sonar Cloud; pre-configurar proyectos rápidamente para su análisis; y visualizar reportes en base a los análisis realizados a lo largo del tiempo. La solución fue validada con una colección de 110 proyectos Java, demostrando que es capaz de disminuir el tiempo de análisis un 92.52% comparando con el análisis manual de Sonar Scanner.

En el futuro consideramos integrar Sonar Juploader con otras herramientas para el análisis estático del código fuente para ampliar el número de métricas generadas, como por ejemplo CKJM, que calcula métricas orientadas no disponibles en Sonar Cloud como cantidad de hijos, profundidad de árbol de herencia, acoplamiento entre objetos, entre otras. Otra alternativa podría ser desarrollar una nueva aplicación que integre el entorno de análisis de CKJM.

## Referencias

1. Parnas, D. L. Some software engineering principles. in Software fundamentals: collected papers by David L. Parnas 257–266 (2001).
2. Irrazabal, E., Vásquez, F., Díaz, R. & Garzás, J. Applying ISO/IEC 12207:2008 with SCRUM and Agile Methods. Commun. Comput. Inf. Sci. 155 CCIS, 169–180 (2011).

3. Lehman, M. M. Laws of software evolution revisited. in *Lecture Notes in Computer Science* (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*) vol. 1149 108–124 (Springer Verlag, 1996).
4. Kitchenham, B. & Pfleeger, S. L. Software quality: the elusive target. *IEEE Softw.* 13, 12–21 (1996).
5. Garvin, D. A. What Does “Product Quality” Really Mean? *MIT Sloan Management Review* 25–43 (1984).
6. Vidal, S. A., Bergel, A., Marcos, C. & Díaz-Pace, J. A. Understanding and addressing exhibitionism in Java empirical research about method accessibility. *Empir. Softw. Eng.* 21, 483–516 (2016).
7. Vidal, S., Bergel, A., Díaz-Pace, J. A. & Marcos, C. Over-exposed classes in Java: An empirical study. *Comput. Lang. Syst. Struct.* 46, 1–19 (2016).
8. Vázquez, H. C., Bergel, A., Vidal, S., Díaz Pace, J. A. & Marcos, C. Slimming javascript applications: An approach for removing unused functions from javascript libraries. *Inf. Softw. Technol.* 107, 18–29 (2019).
9. Tempero, E. et al. The Qualitas Corpus: A curated collection of Java code for empirical studies. in *Proceedings - Asia-Pacific Software Engineering Conference, APSEC* 336–345 (2010). doi:10.1109/APSEC.2010.46.
10. Ceccato, M., Capiluppi, A., Falcarin, P. & Boldyreff, C. A large study on the effect of code obfuscation on the quality of java code. *Empir. Softw. Eng.* 20, 1486–1524 (2015).
11. Okutan, A. & Yıldız, O. T. Software defect prediction using Bayesian networks. *Empir. Softw. Eng.* 19, 154–181 (2014).
12. Behnamghader, P. et al. A scalable and efficient approach for compiling and analyzing commit history. in *International Symposium on Empirical Software Engineering and Measurement* (IEEE Computer Society, 2018). doi:10.1145/3239235.3239237.
13. Carruthers, J. A., Diaz Pace, J. A. & Irrazabal, E. A. A Systematic Mapping Study of Empirical Studies performed with Collections of Software Projects. (2021) <https://arxiv.org/abs/2109.07624>.
14. CKJM - Chidamber and Kemerer Java Metrics, <https://www.spinellis.gr/sw/ckjm>, último acceso 16/09/2021.
15. PMD Source Code Analyzer, <https://pmd.github.io>, último acceso 16/09/2021.
16. Spotbugs, <https://spotbugs.github.io>, último acceso 16/09/2021.
17. Sonar Cloud Documentation, <https://sonarcloud.io/documentation>, último acceso 16/09/2021.
18. Sonar Cloud Sonar Scanner Documentation, <https://sonarcloud.io/documentation>, último acceso 16/09/2021.
19. Corpus-2021 Sonar Cloud Organization, <https://sonarcloud.io/organizations/corpus-2021>, último acceso 16/09/2021.
20. PFC-Test Sonar Cloud Organization, <https://sonarcloud.io/organizations/pfc-test>, último acceso 16/09/2021.
21. Corpus-2021 Github Organization, <https://github.com/Corpus-2021>, último acceso 16/09/2021.

## Storytelling Aplicado al Diseño de Sitios Web: Un Estudio de Caso

Andrea Lezcano Airaldi<sup>1</sup>, David Sandoval<sup>1</sup>, Emanuel Irrazábal<sup>1</sup>

<sup>1</sup> Universidad Nacional del Nordeste, Corrientes, Argentina  
{alezcano, eirrazabal}@exa.unne.edu.ar,  
davidsand2015@hotmail.com

**Resumen.** Las historias son un aspecto ubicuo de la cultura humana; transmiten información de forma memorable y generan compromiso. En los últimos años, el storytelling ha ganado importancia como técnica para contextualizar la información en dominios como el desarrollo de software. El storytelling en el diseño de interfaces web representa un nuevo enfoque para realizar interfaces innovadoras que fomenten la interacción. Los usuarios se sienten atraídos por sitios web interesantes y creativos que ofrecen una experiencia diferente. El objetivo de este artículo es desarrollar e implementar una aplicación web para un grupo de investigación inserto en una institución académica, utilizando una historia como elemento principal para capturar la atención de los usuarios. Mediante la aplicación de técnicas narrativas en el proceso de desarrollo, se logró incorporar la información habitual presente en este tipo de sitios en una historia animada, capturando la atención de los usuarios y manteniendo su interés a lo largo de la navegación.

**Palabras Clave:** Calidad de Software, Narrativa Digital, Experiencia de Usuario.

### 1 Introducción

Las historias son un aspecto ubicuo de la cultura humana; transmiten información en una forma memorable que permite generar compromiso y establecer relaciones[1]. La narración de historias o *storytelling* puede entenderse como la comunicación de ideas, creencias y experiencias a través de historias [2].

En los últimos años, el storytelling ha ganado importancia como técnica para contextualizar la información en dominios comerciales y técnicos, como la gestión del conocimiento [2][3][4] o el desarrollo de software [1][5][6][7]

En la Ingeniería de Software, la creación de prototipos de productos y el storytelling se consideran enfoques distintos pero relacionados para apoyar la obtención y validación de requisitos[8], facilitar la comunicación [9] y la toma de decisiones[10].

En un contexto tan dinámico, el storytelling en el diseño de interfaces web representa un nuevo enfoque para realizar interfaces más atractivas que también aseguren y fomenten la participación de los propios usuarios.

En general, los usuarios se sienten atraídos por sitios web interesantes y creativos que ofrecen una experiencia diferente. Un estudio reciente sugiere que las

funcionalidades no se consideran el único aspecto para satisfacer al usuario [11]. La usabilidad y la eficiencia son otros dos factores que afectan la experiencia del usuario en la web [12]. Los desarrolladores y diseñadores web tienen, por tanto, el desafío de generar compromiso y estimular la curiosidad de los usuarios. La exploración interactiva puede facilitar la comprensión animando a los lectores a construir y responder activamente sus propias preguntas [13][14].

En particular, el uso de sitios web como instrumento de difusión y canal de comunicación es cada vez más relevante entre grupos de investigación, que son un elemento fundamental de las universidades públicas para promover la actividad de I+D y la vinculación con otras entidades a través de su labor diaria. Por otro lado, la efectividad del storytelling a la hora de comunicar temáticas científicas, en particular a audiencias no técnicas, se ha estudiado ampliamente[15], por lo que un grupo de investigación puede valerse de estos beneficios para transmitir la información.

El objetivo de este artículo es desarrollar e implementar una aplicación web para un grupo de investigación inserto en una institución académica, utilizando una historia como elemento principal para capturar la atención de los usuarios. Para ello se adoptó el enfoque de estudio de caso para comprender en profundidad el fenómeno en estudio considerando su contexto real[16].

En primer lugar, se analizaron los sitios web de grupos de investigación de las principales instituciones educativas de la región con el fin de conocerlos y caracterizarlos. Asimismo, se estudió la literatura científica para conocer las prácticas de storytelling aplicadas al desarrollo de software; finalmente, se desarrolló e implementó una aplicación web utilizando técnicas de storytelling e incorporando la información necesaria, característica de grupos de investigación.

Además de esta sección introductoria, el trabajo se encuentra organizado de la siguiente manera. En la Sección 2 se describen los antecedentes y los trabajos relacionados. En la Sección 3 se enumeran las decisiones tomadas para la metodología de investigación propuesta, la recolección y el análisis de los datos, mientras que en la Sección 4 se detallan los resultados y en la Sección 5 se presenta las conclusiones.

## **2 Antecedentes y Trabajos Relacionados**

### **2.1 Grupos de Investigación y Sitios Web Académicos**

Un grupo de investigación es actualmente la principal unidad ejecutora de la ciencia dentro de las universidades. Los investigadores dedican gran parte de su tiempo trabajando con sus colegas más cercanos dentro del grupo de investigación al que pertenecen [17] y estas interacciones generan y explican la mayoría de sus actividades como profesionales de la investigación (líneas de especialización, participación en eventos, elección de revistas donde publicar o dirección de trabajos académicos)

En este sentido, las universidades promueven su actividad de I+D a través de la creación y potenciación de los grupos de investigación que existen en las mismas. La relevancia de un grupo de investigación se percibe a través de su presencia en la red [18] que depende directamente del uso de la web para difundir sus actividades.



Dada la efectividad del storytelling para comunicar la información técnica a audiencias no expertas[15], los grupos de investigación pueden valerse de esta herramienta para transmitir sus hallazgos y enriquecer la experiencia en sus sitios web.

## 2.2 Storytelling en la Web

Un estudio reciente sugiere que las funcionalidades no se consideran el único aspecto para satisfacer al usuario [11]. La usabilidad y la eficiencia son otros dos factores que afectan la experiencia del usuario en la web [12]. Para hacer frente a estas cuestiones, las técnicas de diseño web han sufrido varios cambios en los últimos tiempos, como el uso de páginas estáticas, páginas dinámicas, diseño adaptable o *responsive*, desplazamiento *parallax* [19] y *scrollytelling* [20].

En un contexto tan dinámico, el storytelling en el diseño de interfaces web representa un nuevo enfoque para realizar interfaces más atractivas que también aseguren y fomenten la participación de los propios usuarios.

### *Elementos Básicos de una Historia*

Cada historia tiene una serie de elementos básicos que la hacen un todo[21][22]:

- **El contenido:** el conjunto de ideas y eventos en la historia. En el contexto del sitio web, el contenido es el objetivo que se desea que alcance el usuario y el proceso que lo conduce allí.
- **El tema:** el eje central o concepto de la historia.
- **Los personajes:** vienen en forma de persona, lugar u objeto. Las acciones y pensamientos del personaje guían la historia. Los personajes bien desarrollados ayudan a la audiencia a conectarse con la historia. En un sitio web los personajes pueden ser organizaciones, grupos o empresas, miembros del equipo o usuarios.
- **La atmósfera:** la atmósfera crea la conexión entre la audiencia y la historia. En el diseño web este elemento se logra a través del tono, estilo y colores del diseño.
- **La visual:** el intento del autor de crear una imagen mental (o punto de referencia) en la mente del lector. Las imágenes se pueden utilizar para invocar una respuesta emocional, sensorial (gusto, tacto, olor, etc.) o incluso física. Aplicado a un sitio web, es un componente fundamental en el que influyen fotografías, ilustraciones, tipografía y demás elementos visuales.
- **Lo más destacado:** la parte central de la historia. Causa un punto de inflexión o un cambio en los personajes.
- **El resultado:** viene después del punto culminante y brinda una conclusión. En esta parte, la historia vuelve a la normalidad al disminuir la tensión y la emoción.

Para contar una historia efectiva a través de un sitio web, se deben considerar todos los elementos mencionados anteriormente. En primer lugar, se crea el contenido y el tema, luego se definen los personajes y la atmósfera y se seleccionan las imágenes y los estilos adecuadas. Por último, se incluye un "llamado a la acción": el objetivo que debe alcanzar el usuario [22].

Por otra parte, elementos adicionales como la animación, permiten incluir movimientos, gestos y otros efectos, lo que conduce a una conexión más profunda con la audiencia y una experiencia de usuario enriquecida[13][14].

### 2.2.1 Scrollytelling

El término *scrollytelling* es una combinación de *storytelling* y *scrolling* o desplazamiento. Se utiliza comúnmente para historias complejas que utilizan muchas imágenes y contenido multimedia. Debido a la gran cantidad de componentes individuales, el usuario puede decidir la profundidad de la historia por sí mismo [20].

Los sitios con scrollytelling suelen comenzar con una fotografía o un video a pantalla completa. Al desplazarse, el usuario avanza al siguiente elemento. La estructura narrativa de los sitios que utilizan scrollytelling es lineal o elástica. Una narrativa elástica permite seguir un orden predeterminado. En puntos específicos la historia se bifurca y estas ramas permiten profundizar en la historia. La estructura elástica permite a los usuarios volver al hilo principal de la historia.

Dado que la técnica de scrollytelling suele incluir muchas imágenes y elementos multimedia y se utiliza para contar historias complejas es considerada como un método práctico para agregar información de manera eficiente [23][24].

## 3 Metodología

El objetivo de este artículo es desarrollar e implementar una aplicación web para un grupo de investigación inserto en una institución académica, incorporando estrategias narrativas o de storytelling que permitan incrementar su usabilidad y enriquecer la experiencia de los usuarios.

Para llevar a cabo el estudio se siguieron los pasos propuestos por Yin [16] y enumerados a continuación: diseño del estudio; recolección de datos; recolección de evidencia; análisis de los datos recolectados y reportes de resultados.

### 3.1 Recolección y Análisis de Datos

#### *Análisis de Sitios Web de Grupos de Investigación*

En primer lugar, se accedió al sitio web de las principales instituciones educativas de la región para localizar la información acerca de los grupos de investigación que las componen. De ese modo, se identificó cada uno de los grupos, anotando la URL del sitio web correspondiente en caso de que lo tuviera. Posteriormente, se procedió a navegar de forma manual cada sitio para extraer las secciones que lo componen e información que nos permita caracterizar la presencia de grupos de investigación en la web. Las secciones comunes identificadas se describen la Tabla 1.

**Tabla 1.** Secciones comunes en sitios web de grupos de investigación.

Sección	Descripción
Publicaciones (A)	Trabajos publicados en revistas o congresos.
Quiénes Somos (B)	Información general acerca del grupo.
Miembros (C)	Integrantes que componen el grupo.
Contacto (D)	Información de contacto: formularios, redes sociales, oficina dentro de la universidad, teléfono, etc.
Proyectos (E)	Proyectos en los que participa o participó el grupo.
Líneas De Investigación (F)	Temática de investigación.
Noticias (G)	Novedades relacionadas al grupo o a sus proyectos.
Cursos (H)	Capacitaciones ofrecidas
Patentes (I)	Invencciones (productos o procesos) llevadas a cabo y patentadas por el grupo

#### *Planificación de la Historia*

Una vez concretado este paso, se procedió a planificar la historia, teniendo en cuenta la información presente en sitios web de grupos de investigación, así como también los elementos narrativos.

En base a ello: (i) se describió brevemente la historia mediante casos de uso e historias de usuario y se definió el objetivo a lograr (ii) se desarrolló el storyboard o guion gráfico de la secuencia y (iii) se definió la atmósfera mediante elementos de diseño. Luego de definir las partes básicas, fue necesario determinar el tipo y el ritmo de la animación. Esta decisión es una de las más importantes, ya que se convierte en el hilo conductor de las decisiones del usuario. A su vez, se necesita decidir qué información es más importante o requiere más tiempo para procesar. Para estar seguros de que el usuario pueda seguir la historia y decidir por sí mismo el ritmo de las escenas decidimos utilizar la técnica de scrollytelling.

#### *Cuestionario de Usabilidad*

Para evaluar la usabilidad del sitio web construido se utilizó el cuestionario SUS (System Usability Scale) [25][26], para determinar si los usuarios lograron sus objetivos y tuvieron una experiencia positiva al utilizar el sitio. El mismo consiste en un cuestionario de 10 ítems con cinco opciones de respuesta en la escala de Likert y contempla una variedad de características de usabilidad, como la complejidad del sistema y cualquier soporte que los participantes crean necesario para usarlo de manera efectiva.

## **4 Resultados y Discusión**

### **4.1 Análisis de Sitios Web de Grupos de Investigación**

Se seleccionaron 22 sitios web de grupos de investigación. Los mismos fueron identificados con la letra S y numerados acorde: S01, S02, etc. El nombre completo de estos

grupos, así como la URL de acceso se detallan en el Anexo I<sup>1</sup>. En la Tabla 2 se muestra las secciones identificadas en cada sitio.

**Tabla 2.** Secciones identificadas en cada uno de los sitios web.

	A	B	C	D	E	F	G	H	I
<u>S01</u>	x	x	x	x	x	-	-	-	-
<u>S02</u>	x	-	x	-	-	x	-	-	-
<u>S03</u>	x	-	x	x	x	-	-	-	-
<u>S04</u>	x	x	x	x	x	x	x	-	-
<u>S05</u>	-	x	x	-	x	-	-	x	-
<u>S06</u>	x	x	x	x	-	-	x	x	-
<u>S07</u>	x	x	x	x	x	x	x	-	x
<u>S08</u>	x	x	x	-	x	x	x	x	x
<u>S09</u>	-	x	x	x	x	x	-	-	-
<u>S10</u>	x	x	x	x	x	-	-	-	-
<u>S11</u>	x	x	x	x	x	x	x	x	-
<u>S12</u>	x	x	x	-	-	x	-	-	-
<u>S13</u>	-	x	x	-	-	x	x	-	-
<u>S14</u>	x	-	x	x	-	-	-	-	-
<u>S15</u>	x	-	x	-	x	x	-	-	-
<u>S16</u>	x	-	x	-	x	-	-	-	x
<u>S17</u>	-	x	-	-	x	x	x	-	-
<u>S18</u>	x	x	x	-	x	x	x	-	-
<u>S19</u>	-	x	-	-	-	x	-	-	-
<u>S20</u>	x	-	x	-	x	x	-	-	-
<u>S21</u>	x	x	-	-	x	-	-	-	-
<u>S22</u>	-	x	-	-	x	-	-	-	-

Como puede observarse, las secciones más utilizadas son Miembros (C) y Quiénes Somos (B). Secciones como Proyectos (E) y Líneas de investigación (F) resultan imprescindibles para dar a conocer el contexto del grupo y su área de conocimiento.

---

<sup>1</sup> URL: [https://bit.ly/anexo\\_web](https://bit.ly/anexo_web)

## 4.2 Desarrollo de la Aplicación Web

### *La historia: atmósfera, estilo y personajes*

Para garantizar una alta usabilidad de la aplicación web, se consideraron los usuarios objetivo. En este caso, un usuario objetivo es una persona que desea estar conocer sobre el grupo de investigación, encontrar la información adecuada y potencialmente participar o colaborar.

Para promover la interacción y la curiosidad, en la página de inicio se colocó el logo del grupo con un símbolo que invita al usuario a desplazarse hacia abajo. Como hilo conductor, se optó por la analogía de preparar una taza de café. Las líneas de investigación se representaron mediante “cápsulas” o variedades de café. Al hacer clic en cada una de las líneas, la máquina de café comienza a prepararlo, y a medida que el usuario se desplaza por la página, se observa la caída de café acompañada por información acerca de la línea seleccionada. Esta información se presenta en forma de viñetas de diálogo, con el fin de simular una “conversación de cafetería”, a la vez que narra de manera sencilla los distintos aspectos de la línea de investigación. Estas entidades recibieron el nombre de “artículos”, y el usuario tiene la opción de hacer clic en “continuar leyendo” para profundizar en la historia.

Se decidió mantener esta página lo más simple posible, para no generar distracciones innecesarias o sobrecargas de información. El desplazamiento de la página de inicio finaliza con la taza de café recién preparada, que al rotar toma la forma del logo del grupo, con información de contacto a modo de “llamado a la acción”.

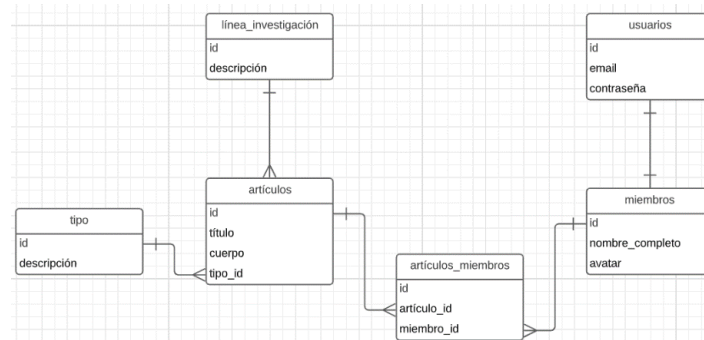
Además de la historia principal, se proporcionó información adicional relativa a cada línea de investigación a través de los artículos. Al hacer clic en “leer más” el usuario es dirigido al artículo completo, donde encontrará detalles sobre los miembros activos en cada línea, trabajos publicados y capacitaciones ofrecidas, entre otros. La Fig. 1 resume los aspectos mencionados anteriormente.



Fig. 1. Capturas de pantalla del resultado final del diseño.

### Implementación

Como entidad principal se definió una “línea de investigación”, la cual contiene artículos. De estos últimos se desprende la información adicional mencionada en la sección anterior. La Fig. 2 muestra el diagrama entidad – relación del sitio.



**Fig. 2.** Diagrama Entidad - Relación del sistema.

La arquitectura del sistema se representó por medio del patrón de diseño Modelo – Vista – Controlador (MVC). Para el desarrollo de la lógica de negocio se utilizó el lenguaje PHP con el framework Laravel y base de datos PostgreSQL; mientras que para la visualización se utilizó HTML, CSS y GSAP [27], una biblioteca JavaScript creada especialmente para animación.

### Cuestionario de Usabilidad

La evaluación se llevó a cabo con 14 participantes. De los cuales 6 eran hombres y 8 mujeres de entre 19 y 34 años (M=26). Solo cuatro participantes tenían experiencia en desarrollo de software.

Preguntas											Puntaje SUS
Part.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	
A	4	2	5	1	4	4	5	1	5	1	85
B	5	2	5	1	3	1	4	1	4	1	87,5
C	1	1	3	1	4	1	3	1	4	1	75
D	2	1	4	1	3	2	4	2	3	2	70
E	5	1	4	2	3	1	3	1	2	1	77,5
F	5	1	4	2	3	1	4	1	4	2	82,5
G	5	3	4	1	4	2	5	2	3	1	80
H	4	1	3	1	4	2	4	1	2	1	77,5
I	3	4	4	1	3	1	5	1	4	1	77,5
J	4	2	5	2	3	1	4	1	5	1	85
K	2	3	3	1	1	2	4	1	4	2	62,5
L	4	1	4	2	4	1	3	2	3	1	77,5
M	2	1	4	1	3	2	4	2	3	2	70
N	1	2	5	1	4	4	5	1	5	1	77,5

**Fig. 3.** Resultados del cuestionario SUS.

La tarea fue navegar a través de la aplicación web y luego completar el cuestionario SUS. Los puntajes se calcularon e interpretaron de acuerdo con el algoritmo de SUS [25]. En general, se observó una experiencia positiva durante la realización de la tarea, con un puntaje mínimo de 62 puntos y un máximo de 87 puntos,  $M=77,5$  (ver Fig. 3).

Esto podría deberse a que se planteó que el “protagonista” de la historia fuera el usuario o la audiencia objetivo, tal y como plantea [22], apelando a la interacción con el sistema para obtener la información. Si el usuario no se desplaza por el sitio, o no selecciona una cápsula, no puede visualizar los detalles de una determinada línea. De esta manera se captura y se mantiene su atención.

## 5 Conclusiones y Trabajos Futuros

Este trabajo explora los beneficios del storytelling en el desarrollo de software. En particular, se enfoca en la creación de una aplicación web para un grupo de investigación inserto en una institución académica.

En el desarrollo web, crear algo que difiera de proyectos similares puede tener un impacto positivo en la experiencia del usuario y en la usabilidad, en especial si el proceso sigue las buenas prácticas y principios conocidos.

Al implementar técnicas de storytelling, se logró incorporar la información habitual en una historia animada, capturando de esta manera la atención de los usuarios y manteniendo su interés a lo largo de la navegación.

Actualmente, existe una brecha de investigación sobre el desarrollo de software y la narración de historias. La literatura existente ve la narración simplemente como la creación de un entendimiento común entre desarrolladores y usuarios, en el sentido de casos de uso e historias de usuario. Como trabajo futuro, se espera profundizar en este aspecto, particularmente cuando se trata de comunicar ideas para la toma de decisiones y transferir conocimientos.

## Referencias

- [1] N. Gershon and W. Page, “What storytelling can do for information visualization,” *Commun. ACM*, vol. 44, no. 8, pp. 31–37, 2001.
- [2] G. Schreyögg, *Knowledge management and narratives: Organizational effectiveness through storytelling*. Erich Schmidt Verlag GmbH & Co KG, 2005.
- [3] E. Meyer, N. A. D. Connell, and J. H. Klein, “A narrative approach to knowledge exchange: an empirical investigation in two contrasting settings,” 2005.
- [4] W. Swap, D. Leonard, and L. A. Mimi Shields, “Using mentoring and storytelling to transfer knowledge in the workplace,” *J. Manag. Inf. Syst.*, 18-1, pp. 95–114, 2001.
- [5] A. Uittenbogaard, “Storytelling for Software Professionals,” *IEEE Softw.*, vol. 30, no. 3, pp. 9–12, 2013.
- [6] E. Wende, G. King, and G. Schwabe, “Exploring Storytelling as a Knowledge Transfer Technique in Offshore Outsourcing,” *ICIS 2014 Proceedings*.
- [7] H. Clausen, “Designing Computer Systems from a Human Perspective: The Use of Narratives,” *Scand. J. Inf. Syst.*, vol. 6, no. 2, Jan. 1994.

- [8] A. Hickey and D. Dean, "Prototyping for requirements elicitation and validation," AMCIS 1998 Proc., p. 268, 1998.
- [9] R. Budde and H. Zullighoven, "Prototyping revisited," in COMPEURO'90, 1990, pp. 418–427.
- [10] J. E. Urban, "Software Prototyping and Requirements Engineering," Rome Lab., 1992.
- [11] M. Kabak, E. Ozceylan, M. Dagdeviren, T. Genc, "Evaluation of distance education website: a hybrid multicriteria approach", Turkish Journal of Electrical Engineering & Computer Sciences, vol.25, pp. 2809-2819, 2017.
- [12] J. Robert and A. Lesage, "From usability to user experience with interactive systems," Handbook of Human-Computer Interaction, pp. 303-332, 2010.
- [13] B. Tversky, J. Heiser, S. Lozano, R. Mackenzie, J. Morrison: Enriching animations. In Learning with Animation (2007), Lowe R., Schnotz W., (Eds.), Cambridge University Press.
- [14] B. Tversky, J. B. Morrison, M. Betrancourt: Animation: Can it facilitate? International Journal of Human-Computer Studies 57, 4 (2002), 247 – 262. URL: <http://www.sciencedirect.com/science/article/pii/S1071581902910177>
- [15] M. F. Dahlstrom, "Using narratives and storytelling to communicate science with nonexpert audiences". doi: <https://doi.org/10.1073/PNAS.1320645111>
- [16] R. K. Yin, Case Study Research Design and Methods. 2003.
- [17] S. Lee, B. Bozeman: The Impact of Research Collaboration on Scientific Productivity. Social Studies of Science. 2005;35(5):673-702. doi:10.1177/0306312705052359
- [18] A. Cabezas-Clavijo, E. Jiménez-Contreras, E. Delgado López-Cózar, "¿Existe relación entre el tamaño del grupo de investigación y su rendimiento científico? Estudio de caso de una universidad española, Revista española de documentación científica, v. 36, 2013.
- [19] F. Wang and S. S. Sundar, "How does parallax scrolling influence user experience? A test of TIME (theory of interactive media effects)", International Journal of Human-Computer Interaction, 2017, pp.1-11 doi: 10.1080/10447318.2017.1373457.
- [20] D. Seyser and M. Zeiller. Scrollytelling – an analysis of visual storytelling in online journalism. In Proc. Int. Conf. Inf. Vis. (IV), 2018.
- [21] D. Jacobs, 2012. Smashing Book #3 1/3: The Extension, The Science Of Storytelling: What Listening To A Story Does To Our Brains. Freiburg : U.S. Magazine, 2012.
- [22] C. Nussbaumer Knaflic, Storytelling with Data: A data visualization guide for business professionals. Hoboken, New Jersey: John Wiley & Sons, Ltd., 2015.
- [23] A. Cairo, The functional art: an introduction to information graphics and visualization. Berkeley, CA: New Riders, 2013.
- [24] J. Lankow, J. Ritchie, and R. Crooks, Infographics: The Power of Visual Storytelling. Hoboken, NJ: Wiley, 2012.
- [25] J. Brooke, 1995: SUS: A quick and dirty usability scale. Usability Eval. Ind.. 189.
- [26] M. Drew, B. Falcone and W. Baccus, 2018. What Does the System Usability Scale (SUS) Measure?. Design, User Experience, and Usability: Theory and Practice, pp.356-366.
- [27] GSAP GreenSock. <https://greensock.com/gsap/>



## Evaluación de la calidad en aplicaciones de realidad virtual: Una revisión sistemática de la literatura

Ledesma, Fabio Alberto<sup>1</sup> y Rambo, Alice Raquel<sup>1</sup>, Syniuk, Rubén Oscar<sup>1</sup>

<sup>1</sup> Universidad Nacional de Misiones, Argentina

[ledesmafabio@gmail.com](mailto:ledesmafabio@gmail.com); [alicerambo@fcegyn.unam.edu.ar](mailto:alicerambo@fcegyn.unam.edu.ar);  
[syniukro@gmail.com](mailto:syniukro@gmail.com)

**Resumen.** La evaluación de la calidad en el desarrollo de software asegura que el producto cumple con los requerimientos y necesidades del cliente. Las aplicaciones móviles de realidad virtual (RV) no deberían estar exentas de ella. Por esto el objetivo del trabajo es conocer los estándares y/o características a tener en cuenta en el desarrollo de aplicaciones móviles con realidad virtual. Para ello se implementó una revisión sistemática de la literatura, a los fines de su identificación. Los resultados obtenidos indican una tendencia a considerar características que mejoran la experiencia de usuario por sobre la calidad de software en general.

**Palabras claves:** Virtual Reality, Quality, Quality Standard, RSL

### 1. Introducción

Esta revisión sistemática de la literatura permite identificar y analizar las características consideradas a la hora de desarrollar aplicaciones móviles con realidad virtual (RV), en 176 estudios publicados entre enero de 2011 y diciembre de 2021. El resultado de este trabajo pretende colaborar con aquellos que necesiten evaluar la calidad en el desarrollo de aplicaciones móviles con realidad virtual.

La calidad es uno de los términos que más se utiliza en la actualidad, si bien abarca un amplio espectro que involucra criterios personales hasta llegar a contemplar la caracterización o evaluación de condiciones en relación a expectativas, especificaciones o pretensiones es necesario establecer métodos medibles o características inherentes. Existen varias propuestas metodológicas enfocadas a la certificación del proceso de desarrollo y a los procedimientos asociados y que inciden en el producto final como ser ISO 9000, CMMI (Integración de sistemas modelos de madurez de capacidades o *Capability Maturity Model Integration*) entre otros. También la ISO/IEC 25000 SQuaRE (*System and Software Quality Requirements and Evaluation*) más orientada a los requisitos y evaluación de calidad de productos software. Sin embargo, surge el interrogante sobre la existencia de un enfoque que se aplique directamente en aplicaciones de realidad virtual, y la necesidad de búsqueda y análisis de la situación referente a la calidad en este tipo de aplicaciones.

### 2. Metodología

Siguiendo las pautas de realización de una RSL propuesta por Kitchenham (2013) Se

identifica el problema de investigación, lo cual permite la planificación y conducción de la revisión. En la etapa de planificación se definen las preguntas de investigación y los criterios de revisión. Luego se identifican los estudios relevantes, se seleccionan los estudios primarios y se evalúan parámetros de calidad establecidos para la revisión.

**2.1. Objetivo:**

Conocer las características para la calidad en el desarrollo de aplicaciones móviles con realidad virtual.

**2.2. Definición del problema:**

¿Existen en la literatura evidencias de evaluación de la calidad en aplicaciones de realidad virtual?

**2.3. Preguntas de Investigación**

Para responder al problema se definen las siguientes preguntas de investigación (PI):

**PI.1:** *¿Cuál es la contribución de los trabajos relacionados a la evaluación de la calidad en aplicaciones de realidad virtual?*

**PI.2:** *¿Existe algún estándar propio para evaluar la calidad en el proceso de desarrollo de aplicaciones RV?*

**PI.3:** *¿Existen antecedentes de parámetros que permitan determinar la calidad de un proyecto de RV?*

**PI.4:** *¿Qué aspectos del proceso de desarrollo de aplicaciones realidad virtual son susceptibles de ser evaluados para asegurar la calidad?*

**PI.5:** *¿Existe trazabilidad entre la aplicación de estándares de calidad y la definición de los parámetros que miden la calidad de un proyecto de RV?*

**2.4. Formulación de las cadenas de búsqueda**

Se establecen un conjunto de palabras claves que permitirán definir las cadenas de búsqueda, como se observa en la Tabla 1. “Palabras claves y Cadenas de Búsqueda”:

**Table 1.** Palabras Claves y Cadenas de búsqueda

Palabras Claves	Cadena de búsqueda generada
Virtual Reality - Quality Assessment - Quality Evaluation - Mobile	“virtual reality” AND (“quality assessment” OR “quality evaluation”) AND “mobile”

Las palabras claves, se presentan en inglés para mejorar la búsqueda debido a que gran parte de la literatura se encuentra en ese idioma, no desestimando la búsqueda en otros idiomas.

**2.5. Selección de las fuentes de datos**

Para el siguiente estudio se tomaron de referencia ocho motores de búsqueda, a saber:

- Repositorio Institucional de la UNLP, Biblioteca SeDiCi, buscando resultados en el ámbito local y preferentemente en idioma español. Recuperado de: <http://sedici.unlp.edu.ar/>
- ACM Library con la intención de obtener resultados más especializados en el ámbito de la informática y computación. recuperado de: <https://dl.acm.org/>
- SpringerLink. Recuperado de: <https://link.springer.com/>
- ScienceDirect. Recuperado de : <https://www.sciencedirect.com/>
- DSpace MIT. Recuperado de: <https://dspace.mit.edu/>
- Harvard Library. Recuperado de: <https://www.sciencedirect.com/>
- Rydalic.org. Recuperado de: <https://www.redalyc.org/>
- IEEEExplore. Recuperado de: <https://ieeexplore.ieee.org/Xplore/home.jsp>

## 2.6. Criterios de selección

A continuación se definen los criterios de selección de estudios primarios. Estos exponen las características aplicadas a la evaluación de calidad de aplicaciones móviles con RV.

### Criterios de Inclusión

**I1:** Artículos que traten los métodos o estándares de evaluación utilizados en aplicaciones móviles de realidad virtual.

**I2:** Artículos publicados en Workshops, Congresos y Revistas.

**I3:** Artículos que combinen dos o más palabras claves.

### Criterios de Exclusión

**E1:** La publicación está fuera del rango temporal definido para la búsqueda. (2011-2021)

**E2:** Artículos incompletos, por estar publicados en páginas que requieren una suscripción arancelada para su acceso total.

**E3:** Artículos privados que requieran del pago de un canon para su acceso total.

**E4:** Artículos que sean publicaciones de tutoriales o sin citas.

**E5:** Libros y capítulos de libros.

**E6:** La aplicación de la palabra clave encontrada en el artículo sea coherente con el tema central.

**E7:** La publicación es un póster.

**E8:** El artículo está publicado en áreas de ciencias no relacionadas a la computación

## 3. Ejecución de la RSL

Una vez realizada la evaluación sobre la planificación de la revisión sistemática, se inició la ejecución de la revisión. Durante esta fase, se realizaron las búsquedas en las fuentes seleccionadas y los estudios obtenidos, en etapas (ver Tabla 2).

### 3.1 Etapa 1

Obtención de fuentes primarias de información: La búsqueda de artículos utilizando el string de búsqueda y tomando todos los resultados obtenidos.

**3.2 Etapa 2: Verificación de Títulos y palabras claves:** Teniendo en cuenta la relevancia de los títulos y palabras claves con relación a las palabras claves establecidas en el string de búsqueda.

**3.3 Etapa 3: Verificación de Abstracts:** Considerando los abstracts frente a las preguntas de inclusión y exclusión establecidas.

**3.4 Etapa 4: Análisis de la calidad de las fuentes primarias:** Considerando la lectura del artículo frente a las preguntas de calidad establecidas.

**3.4.1 Preguntas para el análisis de Calidad:**

**QA1:** El trabajo se adecua al tema de análisis propuesto.

**QA2:** La situación analizada en el proyecto presenta un suficiente desarrollo (al menos más de un caso)

**QA3:** En la propuesta, se presenta suficientes características para medir la “calidad” del producto (al menos más de dos)

**QA4:** ¿Presenta una propuesta desarrollada sobre tecnología emergentes aplicables a una propuesta de tecnología móvil?

**Table 2.** Resultados Definitivos

Fuentes de Datos	ARTICULOS							
	Encontrados	Etapa 1	Etapa 2		Etapa 3			Etapa 4
			Analizados	Dupl	Cumplen (CI)	No Cumplen (CE)	Primarios	
SpringerLink	12	12	2	0	1	1	1	0
ACM	34	1	1	0	0	0	0	0
ScienceDirect	46	46	6	0	2	4	2	2
DSpace MIT	5	5	4	0	1	4	1	1
Harvard Library	50	3	3	1	1	1	1	0
SeDiCI	15	15	12	0	6	6	6	2
Rydalic.org	2	2	0	0	0	0	0	0
IEEEExplore	12	12	8	0	7	1	7	2
TOTAL	176	96	36	1	18	17	18	7

#### 4. Análisis de los resultados:

Algunos de los criterios definidos dentro de cada trabajo están asociados a las características para las cuales fueron creados y su campo de aplicación. Y las características propias de calidad de software se mencionan como consideraciones a ser tenidas en cuenta en las etapas del desarrollo.

- **Duarte, J. P.:** Prioriza la gramática de programación, las interfaces e interacción con el usuario. (Duarte, J. P., 2001)
- **Sgobbi, F., et al:** Consideran importante la interacción, facilidad de uso y la compatibilidad con otros dispositivos. (Sgobbi, F., 2017)
- **Souto Reinheimer, W., et al:** Favorecen la usabilidad en la interacción persona-computadora, la obtención de datos confiables y por sobre todo la diversión del usuario. (Souto Reinheimer, W., 2020)
- **Rodrigues, A. Cheiran J. F.:** Consideran la experiencia de usuario, el nivel de inmersión, realismo y control percibido y los síntomas de discomfort generales. (Rodrigues, A., 2020)
- **Fei, Z., et al:** Analizan los aspectos online (ancho de banda, latencia, pérdida de paquetes) y los offline (calidad, nivel de inmersión, capacidad de giro, interacción y confort). (Fei, Z. 2019)
- **Yu, M. et al:** Hacen hincapié en la experiencia de usuario frente a distintos visores RV. (Yu, M., 2019)
- **Terzié, K. & Hansard M.:** Revisan la incomodidad visual y las características de las imágenes mostradas. (Terzié, K., 2016)

**Tabla 3:** Modelo de calidad ISO 25010<sup>1</sup>

Indicadores	Característica
I-1	Adecuación Funciona
I-2	Eficiencia de Desempeño
I-3	Compatibilidad
I-4	Usabilidad
I-5	Fiabilidad
I-6	Seguridad
I-7	Mantenibilidad
I-8	Portabilidad

En el relevamiento se observa que además de las características propias relacionadas a la calidad de software que algunos trabajos mencionan como consideraciones a ser tenidas en cuenta en las etapas del desarrollo, la importancia al definir indicadores se relacionan más en los resultados de las aplicaciones en el entorno al cual iba destinado. Con la finalidad de determinar si existe alguna trazabilidad entre las características evaluadas en cada trabajo con características propias de la calidad de software es que se propone el uso del estándar ISO/IEC 25010 (ISO/IEC, 2011) Tabla 3, y se realiza un cuadro comparativo Tabla 4 por cada uno de los trabajos evaluados mencionando que características propias de la calidad

<sup>1</sup> ISO/IEC 25010.(01/08/2021). Recuperado de: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

del software estarían contempladas en los indicadores definidos. Teniendo en cuenta las normas ISO/IEC 20510 donde se consideran los 8 indicadores definidos en esta norma, I-1: Adecuación Funcional, I2: Eficiencia de Desempeño, I3: Compatibilidad, I4: Usabilidad, I5: Fiabilidad, I6: Seguridad, I7: Mantenibilidad e I8: Portabilidad.

**Tabla 4:** Trazabilidad de indicadores y papers analizados

Artículos	Indicadores encontrados								
	descripción	I-1	I-2	I-3	I-4	I-5	I-6	I-7	I-8
Duarte, J. P., 2001	Gramática de programación, interfaces, interacción con el usuario				x				
Sgobbi, F., 2017	Interacción, facilidad de uso, compatibilidad con otros dispositivos	x		x	x				
Souto Reinheimer, W., 2020	Diversión, usabilidad en la interacción persona-computadora, obtener datos confiables		x		x				
Rodrigues, A., 2020	Comparación de calidad de interacción en un sitio web de visita virtual desarrollado en WebXR a través de diferentes dispositivos	x	x		x				x
Fei, Z. 2019	Puntaje de sensaciones de no giro y puntaje global para la evaluación subjetiva. Percepción final en realidad virtual transmisión.		x		x	x			x
Yu, M., 2019	Importancia de la experiencia de usuario en el diseño de visores VR		x	x	x	x			
Terzié, K., 2016	Incomodidad visual y características de las imágenes				x				
	Porcentajes de artículos que presentan este indicador	28%	57%	28%	100%	28%	0%	0%	28%

Por lo observado, se puede decir que muchos parámetros definidos de manera personalizada en los artículos pueden encuadrarse dentro de métricas de calidad previstas en los estándares, pero el tipo de aplicación también posee particularidades propias de la RV y específicas del ámbito de aplicación que hacen necesario la definición de indicadores que permiten medir la adecuación de la propuesta al entorno específico.

También se observa que el mayor interés se encuentra en la usabilidad como característica, es decir que en un 100% de los trabajos el producto sea entendible fácil de usar y operar, estéticamente atractivo y accesible. Le sigue, la eficiencia en el desempeño y la utilización de recursos. La pertinencia funcional, la fiabilidad y la portabilidad se encuentran en tercer lugar. Mientras la seguridad y la mantenibilidad no se encuentran mencionadas en los trabajos analizados, lo que no significa que no existan.

## 5. Conclusiones:

En este documento se ha presentado el procedimiento seguido para realizar una Revisión Sistemática de la Literatura, que permitió de manera rigurosa recopilar y analizar la literatura existente sobre las características para la evaluación de calidad de aplicaciones de RV. Se han encontrado 8 estudios primarios publicados entre el 1/1/2011 y 1/02/2021.

Analizando los resultados obtenidos se puede concluir que:

- La calidad de una aplicación de realidad virtual puede estar asociada a diferentes parámetros no excluyentes, como ser: Gráficos (resolución del modelo, imagen o video), Frames por segundos en la ejecución de la aplicación, efectos secundarios en el uso de la aplicación, capacidad de transmisión de datos (para aplicaciones web o multiusuario), eficiencia energética, curva de aprendizaje, grados de libertad, entre otras.
- Se definen parámetros que permiten evaluar cada propuesta basados en las finalidades para las cuales se concibieron cada propuesta, estando orientados a la respuesta o efecto particular de la implementación, pero sin seguir pautas generales sobre el proceso de desarrollo que permitan asegurar que por medio de su seguimiento se obtendrán finalmente los resultados deseados.
- Aunque existen criterios de calidad en la industria respecto al proceso de desarrollo de software, se observan algunas particularidades propias del desarrollo de software RV que llevan a los equipos de trabajo a incluir otras características que les ayuden a determinar la calidad final de sus productos.
- Por lo observado en la Tabla 4 se puede decir que muchos indicadores definidos de manera focalizada en el resultado de los productos presentados en los artículos pueden encuadrarse dentro de métricas de calidad previstas en los estándares, pero el tipo de aplicación también posee particularidades propias de la VR y específicas del ámbito de aplicación que hacen necesario la definición de índices que permiten medir la adecuación de la propuesta al entorno específico.
- También se observa que el mayor interés se encuentra en la usabilidad como característica, es decir que en un 100% de los trabajos el producto sea entendible fácil de usar y operar, estéticamente atractivo y accesible. Le sigue en un 57% la eficiencia en el desempeño y la utilización de recursos. La pertinencia funcional, la fiabilidad y la portabilidad se encuentran en tercer lugar con un 28%. Mientras la seguridad y la mantenibilidad no se encuentran en los trabajos analizados, lo que no significa que no existan.
- Al definir las preguntas de investigación se podría decir que aparecen tres entidades o componentes diferentes a evaluar: entidades de tipo sistema al evaluar aplicaciones de realidad virtual, entidades de tipo proceso al evaluar proceso de desarrollo de aplicaciones RV y finalmente el proyecto de RV. Se pueden tener distintos modelos de calidad, con sus correspondientes

características, propiedades o atributos en cada caso. En este estudio se focalizó la búsqueda en la primera de ellas al buscar características y subcaracterísticas asociadas a aplicaciones (móviles) de realidad virtual.

## Bibliografía

1. B. Kitchenham, P. Brereton. A systematic review of systematic review process research in software engineering. Information and Software Technology archive, Volume 55, Issue 12, pp. 2049-2075, Butter worth-Heinemann Newton, MA, USA. DOI:10.1016/j.infsof.2013.07.010. (2013)
2. Duarte, J. P. Customizing mass housing: a discursive grammar for Siza's Malagueira houses (Doctoral dissertation, Massachusetts Institute of Technology). (2001)
3. Sgobbi, F., Tarouco, L. M. R., & Reategui, E. B. Mundo virtual 3D e Internet das Coisas para motivar mudança de comportamento saudável. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, (19), 7-15. (2017)
4. Souto Reinheimer, W., Carvalho, A., Becker Nunes, F., Duarte Medina, R., & Lopes, V. Uma proposta de diretrizes para fomentar o engajamento dos alunos em ambientes de realidade virtual. TE & ET. (2020)
5. Rodrigues, A., & Cheiran, J. F. P. Virtual look around: interaction quality evaluation for virtual tour in multiple platforms. In 2020 22nd Symposium on Virtual and Augmented Reality (SVR) (pp. 47-56). IEEE. (2020, November)
6. Fei, Z., Wang, F., Wang, J., & Xie, X. Qoe evaluation methods for 360-degree vr video transmission. IEEE Journal of Selected Topics in Signal Processing, 14(1), 78-88. (2019)
7. Yu, M., Zhou, R., Wang, H., & Zhao, W. An evaluation for VR glasses system user experience: The influence factors of interactive operation and motion sickness. Applied ergonomics, 74, 206-213. (2019)
8. Terzić, K., & Hansard, M. Methods for reducing visual discomfort in stereoscopic 3D: A review. Signal Processing: Image Communication, 47, 402-416. (2016)
9. ISO/IEC. ISO/IEC 25010: Software engineering-Software product Quality Requirements and Evaluation (SQuaRE), Quality model. (2011)



## **Análisis y Gestión de Riesgo en Proyectos Software**

### **Un nuevo modelo integrando la metodologías SEI, OCTAVE- S y Magerit3**

Caballero, Ssergio D.<sup>1</sup> y Kuna Horacio D<sup>1</sup>

<sup>1</sup> Instituto de Investigación Desarrollo e Innovación en Informática-  
Facultad de Ciencias Exactas Químicas y Naturales / Universidad Nacional de Misiones  
sergiiodcaballero@gmail.com

**Abstract.** Hablar de riesgos es hablar de futuro, de probabilidades, de incertidumbre, de avances o retrocesos. Toda actividad implica un riesgo, y aunque algunos ubican la etimología en la palabra risco, esto no implica de por sí una valoración negativa, todo cambio implica peligros, pero el peor peligro es la inmovilidad. Un Riesgo es la probabilidad que ocurra una pérdida. Los riesgos técnicos del software son la medida de la probabilidad y severidad de que se produzcan efectos adversos en el desarrollo, adquisición, mantenimiento etc. de sistema[1]. Todas las áreas en el desarrollo de sistemas son fuentes potenciales de riesgos de software. Debido a la importancia de estos, en los proyectos software, se realizará una investigación de distintas metodologías para detectar, analizar, eliminar o minimizar los posibles riesgos a los cuales se somete un proyecto software de pequeña a mediana envergadura durante su ciclo de vida (Planificación, análisis, desarrollo, implementación y mantenimiento) y los riesgos generales a los que se somete el TI<sup>1</sup> de estas organizaciones. Basados en metodologías de análisis y gestión de riesgos estándares de la TI se busca lograr un método ágil, flexible, rápido y sencillo de utilizar por organización que no cuentan con los medios para poder afrontar los altos costos de los estudios de análisis y gestión de riesgos realizados por consultoras privadas o por soporte informático especializado. Y como resultado de la investigación se generará un software prototipo basado en las mejores técnicas de las metodologías investigadas.

**Keywords:** Análisis y Gestión de Riesgos, Auditoría, Ingeniería del Software.

---

<sup>1</sup> Tecnología de la informática

## **1 Desarrollo**

### **1.1 Contexto**

Trabajo Final presentado en la Maestría en Tecnologías de la Información UNNE-UNAM.

### **1.2 Introducción**

El objetivo de este artículo es visibilizar el trabajo final de maestría el cual se basó en generar un modelo de análisis y gestión de riesgos adaptando los requerimientos, las necesidades informáticas y tecnológicas de las organizaciones pyme. Como resultado se obtuvo un modelo simple, ágil y de económica implementación denominado RIC, tomando como base de estudio tres de las metodologías más importantes de análisis y gestión de riesgos en TI[2], utilizando las metodologías SEI<sup>2</sup>, CRM<sup>3</sup>, OCTAVE-S y adaptando técnicas y elementos de la metodología Magerit V3; para facilitar y automatizar la alta carga de trabajo, gestión, control y mantenimiento de proyectos basados en el método creado, en el marco de este trabajo final de maestría se realizó una herramienta software. El modelo y la herramienta sirven para analizar y gestionar los posibles riesgos que pueden tener los activos; generar un procedimiento de seguimiento de los planes de acción de los riesgos gestionados, comunicar y registrar los incidentes ocurridos para evaluar posteriormente el nivel de los ajustes a realizar en el AGR<sup>4</sup>; esto ayudará a eliminar o minimizar los incidentes ocurridos, además, resguardar los activos de las organizaciones para que estos no corran riesgos desconocidos o que su impacto sea mínimo y controlado, la utilización de AGR en los procesos software o de desarrollo de software, aumentan la calidad del proyecto software en su totalidad.

### **1.3 Metodología Propuesta**

Luego de haber realizado el estudio de las metodologías SEI –CRM[3][4], Magerit V3[5], OCTAVE-S[6], se denota que las dos primeras son conocidas por el ámbito profesional y en el contexto del ambiente informático de las pequeñas y medianas empresas en las encuestas analizadas, mientras que OCTAVE-S, está basado en ámbitos de pequeñas organizaciones de no más de 100 personas, e implementa procedimientos sencillos y específicos, apoyándose en procesos como la gestión de los activos vulnerables.

Investigado en detalle las 3 metodologías antes mencionadas y posteriormente al analizar las encuestas realizadas, queda de manifiesto la falta de éxito en la práctica formal y metodológica del AGR.

Como resultante se generó un método adaptado, denominado “RIC”.

---

<sup>2</sup> Software Engineering Institute - Carnegie Mellon University

<sup>3</sup> Continuous Risk Management

<sup>4</sup> Análisis y Gestión de Riesgos

Este método propio, toma los simples pasos de la metodología SEI – CRM, y aplica la base de conocimiento de las amenazas y salvaguardas de la metodología Magerit V3, incorporando como innovación, procedimientos de control de tareas de los planes de acción, gestión de incidentes y la implementación de conceptos de la gestión de conocimiento, con el objetivo de agregar valor al AGR aumentando su conocimiento colectivo dentro de la organización.

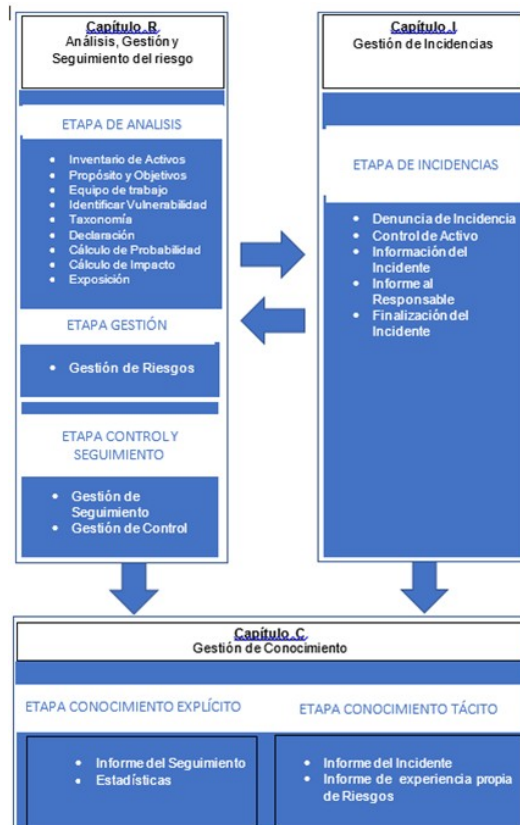
De la metodología SEI se utilizarán las nueve etapas propuestas, de OCTAVE-S la etapa de identificar vulnerabilidades de estructura y de la metodología Magerit V3 las técnicas de clasificación y elementos

El método posee tres etapas externas, el seguimiento de los planes de acción, que es un proceso orientado al control del auditor interno, la gestión de incidencias, y la gestión de conocimiento, que no existe dentro de ninguna de las metodologías antes mencionadas. Ello generará un alto valor agregado al método desarrollado, incorporando técnicas de gestión del conocimiento a la organización, ya que podrán conocer de forma detallada los riesgos a los que se exponen, y los planes de acción y contingencias.

El método tiene como objetivo brindar un marco formal a muchas prácticas realizadas en las pymes aplicando un método simple, intuitivo, que agiliza el reconocimiento de los riesgos, incorporando elementos ya definidos y clasificados por otros métodos o trabajos de investigación, además contribuye ayudar a la organización de las empresas pymes al generar su propia gestión de riesgo, además de aplicar un diseño que se base en estándares, para mejorar y facilitar la auditoría del mismo.

Posee etapas exclusivas para la gestión de incidencias y conocimiento, siempre orientadas a las pequeñas y medianas organizaciones.

La metodología se distribuye en tres capítulos R, I y C cada uno de ellos, poseen Etapas y las etapas poseen [7]



**Fig. 1.** Metodo RIC.

### **1.3.1 Capítulo R - Análisis, Gestión y Seguimiento del Riesgo**

Etapas de Análisis: Se implementan 9 procesos o actividades secuenciales. En los procesos como inventario de activos, propósitos y objetivos e identificar vulnerabilidad se incluyeron subprocesos o tareas. En este capítulo, se evalúan los riesgos de un activo y sus características específicas, la relación entre ellos, la taxonomía y analiza los posibles riesgos, además incorpora las mejores prácticas de las metodologías Magerit v3 y CRM – SEI, y OCTAVE-S, generando una integración propia en este método, mediante formularios sugeridos y simples de administrar.

Etapas de Gestión: Se realiza la actividad de gestión del riesgo, especificando la importancia del riesgo, la información requerida para su seguimiento, el responsable de las tareas de control y los recursos necesarios para llevar dichos controles, además del plan de acción y el plan de contingencia

Etapas de Seguimiento y Control: Esta etapa es creada junto a un plan de seguimiento de salvaguardas, también llamados en este método Plan de Acción, que posee actividades de seguimiento y control para cada riesgo que se gestione, designa también a una persona que realiza las tareas de seguimiento y control, con el fin de que los elementos del plan de acción estén en perfecto orden y control.

### **1.3.2 Capítulo I – Gestión de Incidencias**

Etapas de Incidencias: Cuando ocurre un problema hay que buscar una solución rápida y práctica, para reducirlo o controlarlo. La característica de esta etapa es, el control y gestión de las incidencias ocurridas, es una etapa creada exclusivamente para este método ya que la gestión de incidentes no existe en otras metodologías de AGR. Dicha gestión posee la particularidad de registrar los activos afectados, estén o no en el AGR, el tiempo transcurrido ante la declaración del incidente y su culminación, la probable solución existente, y la designación de un responsable, a cargo de la solución del problema. Como valor a la gestión de incidentes, se puede divisar los activos que tuvieron incidentes y no fueron gestionados, detallando las tareas para la resolución de problemas, que podría resultar útil como información real de los riesgos en la etapa de análisis de riesgo.

### 1.3.3 Capítulo C, Análisis, Gestión de Conocimiento

Etapas Conocimiento Explícito: El conocimiento se incrementa utilizando métodos específicos de la comunicación, con datos cuantitativos y parametrizables

Etapas Conocimiento Tácito: Incorpora conocimiento cualitativo, en lenguaje natural, para obtener conocimiento desestructurado.

### 1.4 Validación

Se realizó la validación del modelo a través de la herramienta SCRisk creada en el trabajo final de maestría, la misma, se realizó en un ámbito real, fue instalada en el Servicio Médico Asistencial de la Universidad Nacional de Misiones, y como resultado se pudo observar como enseñanza:

Característica	Descripción	Efecto
POSITIVO	Relevamiento de Activos e identificación de tipos y dimensiones	Se notó una valoración de los activos, y mayor conocimiento en base a su tipo y dimensión. Se utilizó la experimentación para generar un listado de activos tecnológicos.
POSITIVO	Definición de proyectos	Se observó la importancia de definir el propósito y objetivo del proyecto. Esto llevó al personal a reunirse con la gerencia e informar la importancia del AGR.
POSITIVO	Generación automática de Taxonomía	La herramienta genera automáticamente la taxonomía de los riesgos por proyecto, ahorrando tiempo de trabajo del personal.
NEGATIVO	Definición de consecuencia y efecto en la descripción del riesgo.	La definición de la consecuencia y el efecto, trajo muchas dudas al personal, por no tener una clara visión de la diferencia de los conceptos.
NEGATIVO	Tablas de comparación de probabilidad por proyecto	Se observó que diferenciar las tablas comparativas de probabilidad de ocurrencia por proyecto es innecesaria, debido a que dicha tabla se puede utilizar como referencia para cualquier proyecto.
POSITIVO	Fuente de información de la gestión de Riesgo	Establecer un listado de fuentes de información que se puedan utilizar en la declaración de cualquier

		riesgo, esto minimiza la repetición del dato.
POSITIVO	Generación de plan de acción automático	Generar un plan de acción automático, brinda ahorro de trabajo al personal y minimiza la posibilidad de saltar algún paso en la confección del mismo.
NEGATIVO	Identificar la amenaza en la incidencia	En la gestión de incidencia, se solicita identificar la amenaza por la cual se generó la misma, y el personal actuante puede no tener conocimiento de las amenazas que pueden afectar a los activos.

En la experimentación hemos observado algunas falencias por parte del método que se deberán discutir en futuras versiones, pero, observamos características positivas que generan tanto el método como la herramienta que proveen de valor a la organización, ahorra trabajo del personal, y visualizan la importancia de los activos y del proceso de gestión de riesgos.

Para la validación del modelo, se utilizó el método basado en juicios de expertos, utilizando como base la escala de tipo likert<sup>5</sup> en el cual se generó una encuesta basada preguntas.

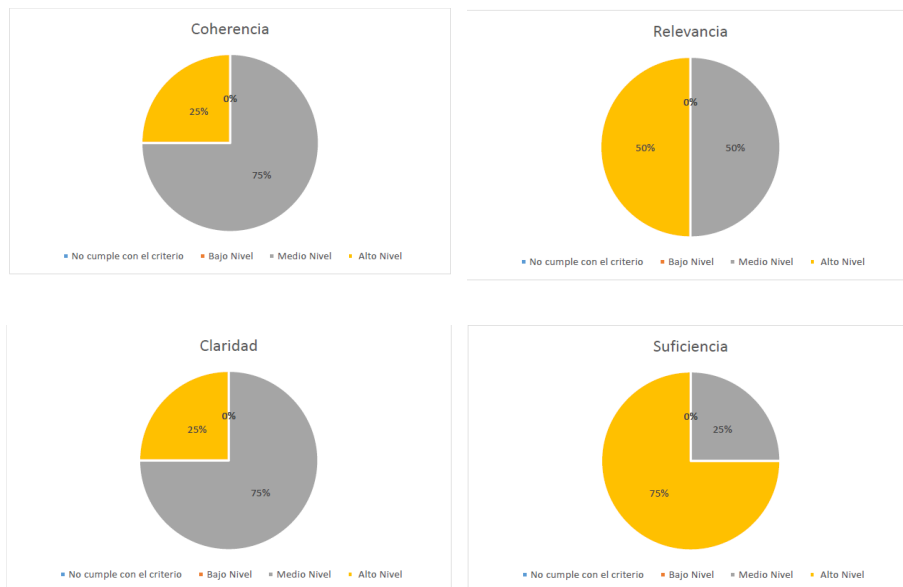
A partir de la selección de cuatro expertos, se entregó un email con la invitación personalizada, explicando los objetivos de la prueba, el propósito de la validación y demás especificaciones para contextualizar a los jueces; así como se adjuntaron las 4 preguntas y la unidad del método RIC, el tiempo límite de respuesta fue un lapso de una semana. Se realizó el método denominado “agregación individual”, en la que se obtiene la información de cada uno de los expertos, de manera individual, sin que éstos se encuentren en contacto, de forma anónima y confidencial, por lo cual, los mismo fueron seleccionados de distintas industrias de las tecnologías informáticas.

Las preguntas a contestar poseían las mismas posibles respuestas : 1- No cumple con el criterio, 2- Bajo Nivel, 3- Medio Nivel, 4- Alto Nivel, y los parámetros a evaluar al estudio del método fueron:

- Coherencia
- Relevancia
- Claridad
- Suficiencia

---

<sup>5</sup> Escala psicométrica comúnmente utilizada en las investigaciones de ciencias sociales que emplean cuestionarios.



### 1.5 Resultado Obtenidos

Se pudo observar en la bibliografía relevada y en otros trabajos de investigación, que una gran parte de organizaciones han modificado el método original del análisis y gestión de riesgos adaptándolo a sus necesidades, pero no se observó, la adaptación de un método, utilizando elementos de otro método, es decir, integrar características de diferentes metodologías, destacándose la posibilidad de que puede ser utilizada por cualquier tipo organización pyme.

Métodos de AGR hay muchos, y a diferencia de los relevados, este método busca la agilidad y adaptabilidad de las actividades, impulsa a tratar únicamente las tareas necesarias.

Para gestionar la TI de la organización, no es suficiente, solo con la implementación de un método de AGR formal, por lo que, se propone una instancia de generación de tareas de control de los riesgos a gestionar, sumando una agenda de actividades de control.

La realidad nos demuestra que ocurren incidentes en los activos de TI, y que no todos son gestionados, por ese motivo se creó la fase de incidencias, que permite una gestión de la totalidad de los incidentes de los activos, resolviendo e informando ese incidente; teniendo en cuenta que alguno de ellos forman parte de un proyecto de AGR, otros fueron analizados pero no gestionados, y que existen activos que no se incluyen en proyecto alguno, el método clasifica los activos y riesgos que participaron en el incidente.

Cuando un activo tiene un incidente y el riesgo ejecutado es gestionado, el módulo propone un análisis del plan de contingencia, para constatar que es correcto, y de no serlo, propone ajustar dicho plan. Cuando No es gestionado, aumenta la probabilidad de ocurrencia del riesgo. Además, se suma a la base de conocimiento, el plan de contingencia o tareas ejecutadas para solucionar la incidencia, que puede ser utilizada en proyectos de análisis y gestión de riesgos futuros, restando carga de trabajo al recurso humano.

Comunicar los riesgos que contemplan los activos de TI mejora el conocimiento de valor de estos. Que la comunicación sea gestionada y información se establezca como activo de la organización, son valores buscados por toda pyme.

A fin de cumplir con la totalidad de los objetivos propuestos e independizar el método de la herramienta, facilitando así el trabajo de los auditores informáticas, el método incluye formularios.

Sitio donde se encuentra el Trabajo final de Maestría:

[https://www.fceqyn.unam.edu.ar/mti/wp-content/uploads/TFM\\_CaballeroSergio.pdf](https://www.fceqyn.unam.edu.ar/mti/wp-content/uploads/TFM_CaballeroSergio.pdf)

### **1.5 Formación de recursos humanos**

Tesis de Maestría en Tecnología de la Información UNNE – UNaM.

Trabajo de Investigación "Análisis de Riesgo en Proyectos Software adaptados a la realidad tecnológica y Socio económica de la Pcia. de Misiones " – Secretaría de Investigación y Post Grado – FCEQyN

## **References**

- [1] M. J. Butkovic et al., "CERT ® Resilience Management Model , Version 1 . 2," Sei, no. November, pp. 1–520, 2010, doi: CMU/SEI-2010-TR-033 ESC-TR-2010-033.
- [2] L. Jaunarena and C. Belletti, "Uso de métricas para la gestión de riesgos," 2002, Accessed: Mar. 20, 2018. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/23022>.
- [3] S. D. Caballero, "Un nuevo método integrando la metodología SEI y Magerit2," Universidad Nacional de Misiones, 2010.
- [4] S. Maniasi, P. Britos, and R. García-Martínez Global, "A Taxonomy-Based Model for Identifying Risks," Proc. V Ibero-American Symp. Softw. Eng., no. March, pp. 13–18, 2006.
- [5] M. A. Amutio Gómez, MAGERIT – versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Libro I - Método. 2012.
- [6] O. Páez Obando, "Aplicación de la norama OCTAVE-S en la Empresa Pirámide Digital CIA. LTDA," PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR, 2013.
- [7] S. D. Caballero, "Trabajo Final de Maestría en Tecnología de la Información Análisis y Gestión de Riesgo en Proyectos Software Un nuevo modelo integrando la metodología SEI y Magerit3 Caso de estudio Servicio Médico Asistencial de la Universidad Nacional de Misiones," 2021, [Online]. Available: [https://www.fceqyn.unam.edu.ar/mti/wp-content/uploads/TFM\\_CaballeroSergio.pdf](https://www.fceqyn.unam.edu.ar/mti/wp-content/uploads/TFM_CaballeroSergio.pdf).





## i-QuAGI: Un enfoque inteligente para la gestión de calidad en proyectos de software ágiles

Jazmín Teng, Lucas Maidana, Nicolás Tortosa, Noelia Pinto, César J. Acuña

<sup>1</sup> Centro de Investigación Aplicada a TIC, CInApTIC  
French 414, Resistencia, Chaco, Argentina

{jazminteng, lucasmaidanasoy, nicotortosa, ns.pinto,  
csr.acn}@gmail.com

**Abstract.** En la actualidad, el conocimiento constituye un recurso realmente competitivo de las empresas. De hecho, muchas de ellas evalúan la calidad de sus procesos teniendo en cuenta factores como la eficiencia en la gestión del conocimiento. Por lo tanto, una manera de contribuir con las organizaciones en esto, es diseñar sistemas que faciliten el seguimiento de calidad de sus proyectos desde la gestión del conocimiento generado. Así, y a partir de experiencias anteriores de vinculación con diversas organizaciones, se presenta en este trabajo, una primera aproximación al diseño de i-QuAGI, un enfoque inteligente que se incorpora al framework AQF y que permite liberar del trabajo manual a los equipos, ofreciendo recomendaciones automáticas que permitan favorecer la calidad del proceso de desarrollo de software, cuando es guiado por enfoques ágiles.

**Palabras clave:** Calidad de Software, Enfoques ágiles de desarrollo de software, Sistemas Multi-agente, Proyectos ágiles de software.

### 1 Introducción

La Ingeniería de Software Inteligente (ISE) se refiere a la aplicación de técnicas inteligentes a la ingeniería de software [1]. Se define una "técnica inteligente" como una técnica que explora datos (de artefactos digitales o expertos en el dominio) para el descubrimiento de conocimientos, el razonamiento, el aprendizaje, la planificación, el procesamiento del lenguaje natural, la percepción o la toma de decisiones de apoyo.

Es posible utilizar técnicas de inteligencia artificial (IA) para ayudar a construir productos de software, sin embargo, el objetivo de la ISE no es ese, sino que esta disciplina se ocupa principalmente de la definición de procesos que permitan incorporar técnicas de IA en los sistemas resultantes [2].

En el escenario del desarrollo de productos de software, una variable fundamental es la gestión del conocimiento generado. De hecho, los equipos responsables deben tener, no solo la experiencia técnica en el campo, sino el conocimiento pertinente para aplicar cualquiera de las metodologías que la Ingeniería de Software demanda para ser aplicada de la mejor manera. La falta de procedimientos y herramientas formales en la

gestión de conocimiento, en el desarrollo de software, impiden su acertada ejecución generando diversos problemas que redundan en la calidad del producto entregado [3]. Asociado a esto, se destaca la popularidad ganada por el uso de enfoques ágiles [4], pues ofrece a los equipos el control de requerimientos variables, la gestión efectiva y eficaz de los grupos de trabajo y el involucramiento y empoderamiento del cliente dentro del proyecto.

Desde hace unos años, resulta imperante la necesidad de la gestión de calidad de los procesos en equipos dedicados al desarrollo de proyectos ágiles<sup>1</sup>, por lo que, producto de investigaciones anteriores, se ha presentado el framework AQF [5][6], una propuesta que integra un modelo de calidad (QuAM) junto a una herramienta de software (QuAGI) que permite la automatización de dicho modelo y que ayuda a la gestión de este tipo de proyectos.

En función a resultados obtenidos luego de diversas experiencias de validación llevadas a cabo en ambientes reales, y si bien los equipos participantes han manifestado adaptarse fácilmente a la herramienta de software QuAGI, se ha observado que resulta necesario enriquecer el framework de forma tal de ofrecer una nueva herramienta que permita liberar de trabajo de monitorización manual de los proyectos

Derivado de esta necesidad detectada, en este artículo se presenta una evolución del framework AQF a partir de la incorporación de una primera aproximación al diseño de i-QuAGI, un enfoque inteligente que permita recomendar acciones al equipo de forma tal de mejorar los niveles de calidad del ciclo de desarrollo ajustando los factores que sean necesarios. Se busca, entonces, incorporar al framework AQF una herramienta que dé soporte al equipo de desarrollo, a partir de recomendaciones automáticas que surjan del seguimiento del proyecto ágil y sus actividades, las cuales muchas veces son afectadas por acciones en segundo plano que pasan desapercibidas e impactan negativamente en los niveles de calidad del proceso de desarrollo asociado.

El resto del artículo se estructura de la siguiente forma: la sección 2 describe los trabajos relacionados a la temática abordada; la sección 3 presenta brevemente una primera aproximación al diseño de i-QuAGI; por último, la sección 4 resume las conclusiones a las que se llegó durante la realización de este trabajo y se presentan futuras líneas de investigación.

## 2 Trabajos Relacionados

En el campo de la ingeniería de software son muchas las investigaciones que se centran en presentar propuestas sobre sistemas inteligentes para el seguimiento de proyectos de software. Por un lado, una mayoría importante de contribuciones se corresponden con “Aplicaciones prácticas”, propuestas que describen soluciones de aplicación práctica dentro del contexto de una organización en particular. Enfocadas en la estimación del esfuerzo cuando el ciclo es guiado por enfoques ágiles. Algunas de estas propuestas se presentan, brevemente, a continuación.

---

<sup>1</sup> Se utiliza el término “proyectos ágiles” para referenciar proyectos de software cuyos ciclos de desarrollo estén guiados por enfoques ágiles

En referencia a esto último, el trabajo presentado en [7] propone un enfoque que busca mejorar la estimación del esfuerzo del software y la gestión del conocimiento de los proyectos de software al centrarse en el proceso y las prácticas de Scrum utilizando el modelo de ontología en un sistema de estimación de agentes múltiples. La aplicación motiva, además, a guardar regularmente un conocimiento tácito significativo de situaciones únicas en forma de lecciones aprendidas durante el desarrollo del proyecto. En esta propuesta, los agentes del sistema acceden a la base de conocimientos existente y realizan de manera autónoma sus actividades de inferencia utilizando la lógica de descripción según los requisitos especificados por el Scrum Master y le responden con una estimación adecuada en forma de tiempo, recursos y lecciones aprendidas para el éxito de proyectos futuros.

Otro ejemplo de aplicación práctica se expone en [8], donde a partir del uso de técnicas de aprendizaje automático para modelar los tipos de visualización existentes con las historias de usuario correspondientes, se diseña e implementa un prototipo de sistema de recomendación ReVizy para predecir el tipo de visualización recomendado para nuevas historias de usuario.

En el caso del trabajo referenciado en [9], se presenta el desarrollo de un framework que describe la estimación de un desarrollo de software en las etapas iniciales y consta de enfoques de retroalimentación y retroalimentación en todo el ciclo de desarrollo, utilizando para ello técnicas basadas en Redes Bayesianas. El artículo concluye con la implementación de la técnica de estimación mediante la implementación de un pequeño proyecto.

Una propuesta diferente se presenta en [10], donde sus autores comparten un sistema inteligente basado en algoritmos genéticos para resolver el problema de estimación del esfuerzo de desarrollo de software en proyectos ágiles. El trabajo se basa en la morfología matemática, que consiste en una neurona híbrida-artificial (perceptrón Dilation Erosion) extendido desde el concepto de teoría de la red completa, sus autores modifican los modelos existentes realizando adaptaciones al enfoque ágil para la estimación en proyectos de software.

En [11], se propone un sistema inteligente de recomendación y apoyo a la toma de decisiones que da soporte al Scrum Master para estimar mejor un próximo proyecto de software en términos de costo, tiempo y recomendaciones de recursos humanos. La aplicación toma como base las mejores prácticas recolectadas de casos exitosos y datos históricos guardados.

Otro trabajo relacionado que se destaca es el caso de la propuesta presentada en [12], donde se describe un bot de refactorización de software inteligente, llamado RefBot. Integrado en el sistema de control de versiones (por ejemplo, GitHub), el bot supervisa continuamente el repositorio de software y reacciona frente a determinados eventos del ambiente.

Por otro lado, se diferencian también las propuestas que se enfocan en hacer de soporte o gestionar diferentes prácticas ágiles independientemente del enfoque que caracterice al proyecto. Así, un caso, se representa en el estudio referenciado en [13], el cual describe una solución llamada Sponto, una aplicación para automatizar la producción de historias de usuario en base a ontologías, ofreciendo plantillas tanto a desarrolladores como a la comunidad que desee reutilizarlas. Si bien no se focaliza en un único enfoque

ágil, ni en todo el ciclo de desarrollo, esta propuesta se destaca por la posibilidad de reducir el tiempo empleado en recrear artefactos ya utilizados en proyectos de similares características. La herramienta no solo favorece la producción de historias de usuario, sino que las utiliza como entradas para obtener otros artefactos útiles para el desarrollo de software (por ejemplo, scripts de base de datos, diagramas de Business Process Model (BPM), fragmentos de código JAVA, entre otros).

Otra aplicación que considera también la elicitación de requerimientos en proyectos ágiles de software, es la presentada en [14], donde sus autores proponen una ontología de dominio para dar soporte, en contextos de Scrum, al Product Owner durante el proceso de ingeniería de requisitos en la gestión ágil de proyectos de software. La fortaleza de la propuesta radica en que integra la ontología de soporte de información para el proceso de ingeniería de requisitos en la gestión ágil de proyectos y la ontología de dominio de aplicación del producto software que se está desarrollando.

Sin embargo, la mayoría de estas herramientas fallan a la hora de dar soporte a los procesos de toma de decisiones y no ofrecen asistencia en cuanto a identificar qué prácticas o acciones llevadas adelante en la gestión de proyectos de software están afectando negativamente a la calidad del proceso, repercutiendo directamente en la calidad del producto final de software. Esto último se debe a que la mayoría se abocan al seguimiento de un único proyecto y no se detienen en recopilar experiencias de proyectos anteriores, desaprovechando de esta manera todo el conocimiento previamente adquirido. Normalmente, esta experiencia es sólo adquirida por el equipo de desarrollo y cuando éste ya no existe, la información que no esté asociada a la herramienta de seguimiento de proyectos y que se relacione al equipo en sí mismo, se pierde. Esto termina perjudicando a la organización [15] y derivando en lo que se conoce como Amnesia Organizacional (AO).

El artículo que aquí se presenta, expone la primera aproximación al diseño de un enfoque inteligente, que se constituye como una nueva herramienta integrada al framework AQF, con el objetivo de ofrecer una alternativa que permita la gestión del conocimiento y la toma de decisiones en el equipo de modo de evitar realizar acciones que repercutan negativamente en el proyecto de desarrollo basado en enfoques ágiles.

### **3 Primera aproximación al diseño de i-QuAGI**

La herramienta i-QuAGI, representa un sistema multiagente (SMA), compuesto por agentes de software (AS)<sup>2</sup> donde el control del sistema es distribuido, y en el que los agentes comparten un modelo de comunicación, caracterizado por reacciones ante eventos generados a partir del uso de QuAGI por el equipo del proyecto de software y que puedan afectar a la calidad final del proceso ágil subyacente.

Para modelar el SMA, desde su diseño y hasta su implementación, el equipo de trabajo se abocó a obtener los diagramas establecidos en la metodología Agile PASSI [16]. Esta es una versión que, según expresan sus autores, incorpora la filosofía de enfoques

---

<sup>2</sup> Se corresponde con la descripción de una entidad de software capaz de actuar con cierto grado de autonomía para completar tareas complejas

ágiles de desarrollo de software a la metodología de Proceso para la Especificación e Implementación de Sociedades de Agentes (PASSI por sus siglas en inglés) [17].

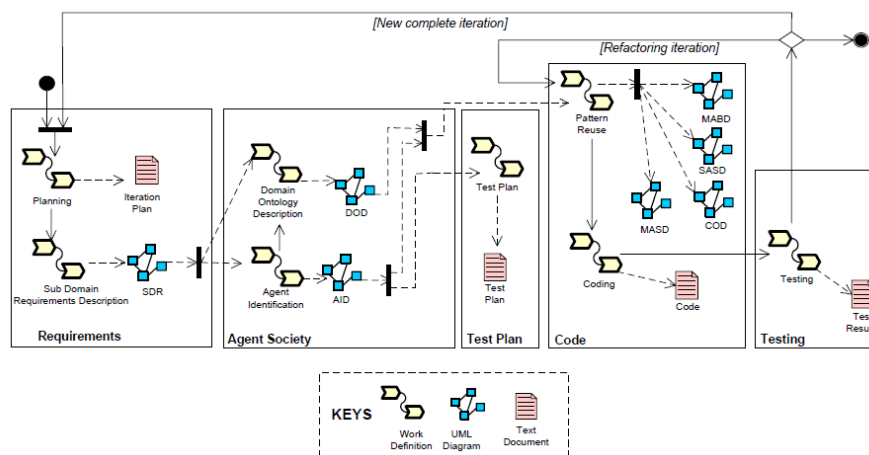
La selección de la metodología como herramienta para el diseño de i-QuAGI se basó fundamentalmente en 3 aspectos:

- Agile PASSI se caracteriza por un proceso iterativo y, además, dado que la implementación se encuentra explícitamente dentro del proceso de la metodología, se dan actividades de testing entre iteraciones que permite validar las versiones que se obtienen de cada modelo.
- Agile PASSI es una metodología paso a paso que abarca desde los requerimientos hasta el código para el diseño y el desarrollo de los SMA.
- Agile PASSI logra integrar modelos y conceptos que provienen de la ingeniería orientada a objetos y de la inteligencia artificial basados en la notación de UML, con la cual el equipo de investigación abocado a este proyecto tiene familiaridad.

La metodología Agile PASSI distingue 4 fases:

- Requerimientos del Dominio (Requirements, en inglés): un modelo de los requisitos del sistema que se compone de dos pasos, Planificación y Descripción de requisitos del dominio.
- Sociedad de Agentes (Agent Society, en inglés): visión de los agentes implicados en la solución, sus interacciones y sus conocimientos sobre el mundo.
- Implementación de código (Code, en inglés): un modelo de dominio de solución a nivel de código.
- Pruebas (Testing, en inglés): descompuesto en el plan de prueba antes de la implementación y la ejecución de pruebas a continuación de ello.

La interacción entre fases y los artefactos que caracterizan a cada una de ellas se pueden observar en la Figura 1, que se expone seguidamente.



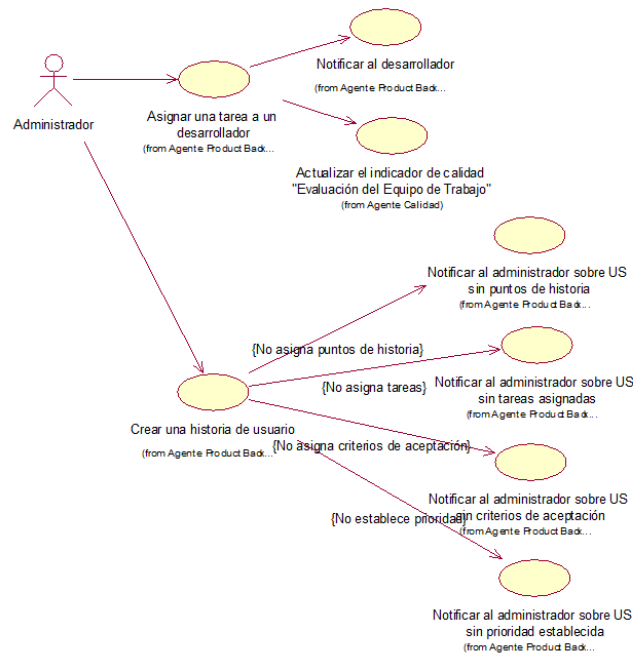
**Fig. 1.** Proceso Agile PASSI [17]

En base a lo anteriormente referenciado, a continuación, se presenta el avance del proyecto i-QuAGI en sus etapas iniciales de modelado.

### 3.1 Modelo de Requerimientos de Dominio

El Modelo de Requerimientos de Dominio, representa el primero de los 4 propuestos por la metodología PASSI y está compuesto por 2 etapas: Planificación, a partir de la *Descripción del dominio*, y Descripción de Requerimientos a través de los modelos de *Identificación de agentes*, *Identificación de rol* y *Especificación de tarea*.

Haciendo foco en el modelo correspondiente a la *Descripción del Dominio*, éste consiste en expresar los requerimientos funcionales del sistema que se diseña, en términos de diagrama de casos de uso. Actualmente, el equipo se encuentra trabajando en la especificación de los requerimientos del agente Product Backlog, identificando las relaciones con los eventos disparados por el actor Administrador en un contexto de seguimiento de proyecto ágil. De acuerdo a los eventos y las acciones identificadas, se logró expresar dichos requerimientos en el diagrama de caso de uso que se muestra en la Figura 2.



**Fig. 2.** Esquema inicial de un Diagrama de Descripción del Dominio

A partir de este primer diagrama, y con la ayuda de PASSI Toolkit, se inició con la fase de Requerimientos del dominio, modelando el componente de: *Identificación del Agente*. En este caso, se agruparon las funcionalidades de un agente en un paquete de

caso de uso<sup>3</sup>. Como resultado de este proceso se obtuvo un Diagrama de Identificación de Agentes que se presenta en la Figura 3, donde es posible identificar las funcionalidades y responsabilidades de cada agente. Las entidades externas que interactúan con el sistema se representan como *actores*, que, en este caso, corresponde al rol Administrador.

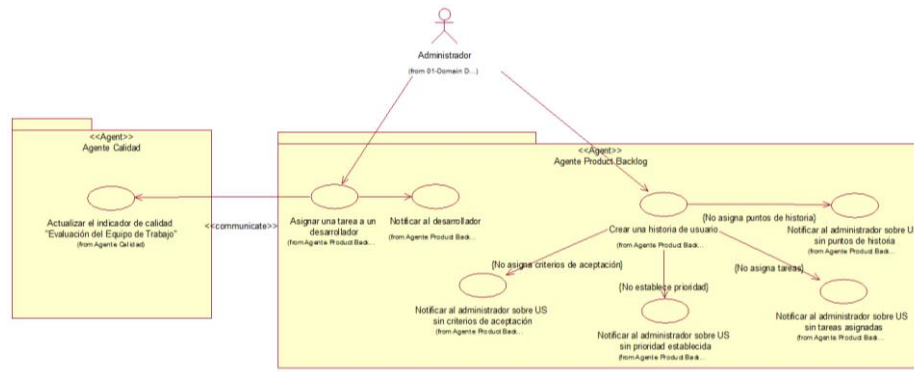


Fig. 3. Esquema inicial de un Diagrama de Identificación de Agentes

Con el modelo anterior obtenido, el equipo se abocó a la obtención del modelo correspondiente a la *Identificación de Roles*. Según la metodología Agile PASSI, el rol es una función que asume temporalmente el agente en su contexto mientras persigue un sub-objetivo. Esto significa que un agente puede ocuparse de varios roles funcionales para cumplir con su objetivo.

En este trabajo, se presenta en la Figura 4, una primera instancia ejemplo de comunicación entre los roles identificados en i-QuAGI, a través de un diagrama de secuencia: Agente Product Backlog y Agente Calidad.

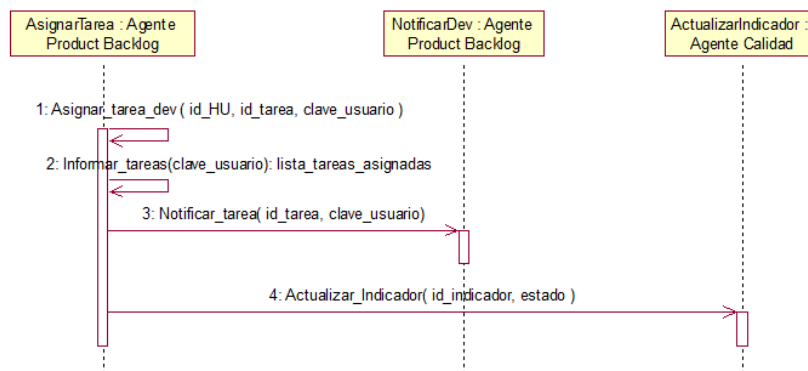


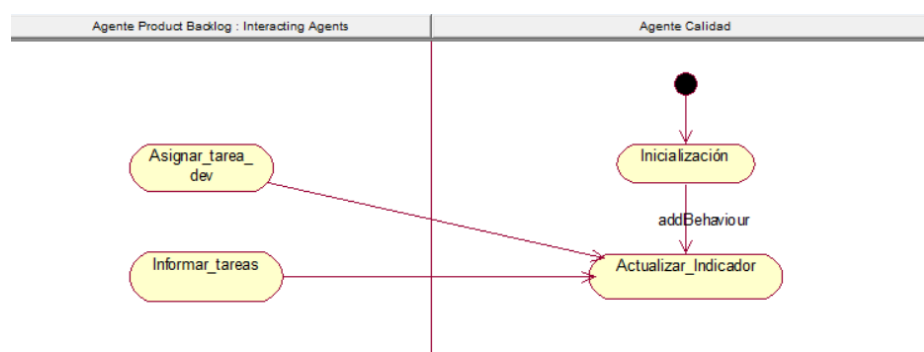
Fig. 4. Esquema inicial de un Diagrama de Identificación de Roles

<sup>3</sup> Para lograr mejor comprensión, se subdivide el diagrama de casos de uso en paquetes. Los paquetes de casos de uso son la forma de agrupar a estos últimos según algún criterio [18]



Para completar el modelo de Requerimientos del Dominio, se ha trabajado en el componente de *Especificación de Tarea*, actividad que consiste en la especificación de las capacidades de cada agente mediante diagramas de actividad.

En cada diagrama de actividad correspondiente al componente Especificación de Tarea, se dividen, por un lado, las tareas de un agente en específico y, por otro lado, las tareas de los agentes que interactúan con él. Por ejemplo, en la Figura 5 se incluye el caso que modela la interacción entre el Agente Calidad que, al recibir datos y detectar el evento de asignación de tareas, debe actualizar indicadores del informe de calidad correspondiente.



**Fig. 5.** Esquema inicial de un Diagrama de Especificación de Tarea para el agente Calidad

Si bien aún el Modelo de Requerimientos del Dominio del SMA i-QuAGI se encuentra en revisión, el equipo ha comenzado a organizar y planificar el trabajo correspondiente a la próxima fase del diseño del sistema: Sociedad de Agentes. El objetivo principal de esta etapa se centra en definir una solución orientada a agentes del problema, no solo identificando a cada uno de ellos, sino que estableciendo las interacciones y las dependencias existentes entre los dos.

## 4 Conclusiones y Trabajos Futuros

En este artículo se ha presentado una aproximación al diseño de un enfoque inteligente que permita recomendar acciones a un equipo de desarrollo de software ágil, de forma tal de mejorar los niveles de calidad del proceso ajustando los factores que sean necesarios. Para ello se ha descrito el avance logrado para la obtención del sistema multi-agente, usando la metodología Agile PASSI y exponiendo cada uno de los modelos obtenidos en las fases de diseño del sistema.

El sistema i-QuAGI, se pretende incorporar a una nueva versión del framework AQF, como una herramienta que dé soporte al equipo de desarrollo, a partir de recomendaciones automáticas que surjan del seguimiento del proyecto ágil y sus actividades, las cuales muchas veces son afectadas por acciones en segundo plano que pasan

desapercibidas e impactan negativamente en los niveles de calidad del proceso de desarrollo asociado. Por esto, resulta fundamental el modelado que permita el diseño y revisión del sistema multi-agente desde sus requerimientos hasta su validación.

Como trabajos futuros se prevé continuar desarrollando las fases siguientes, según lo establecido por la metodología Agile PASSI, e iniciar el diseño de estrategias de validación de los modelos que se vayan logrando en cada etapa, a fin de ajustar y mejorar los aspectos que sean necesarios en cada instancia. Esto, a su vez, seguirá enriqueciendo el modelo QuAM y la plataforma QuAGI, de forma tal de añadir modificaciones que contribuyan a una mejor versión del framework AQF.

## 5 Agradecimientos

Este trabajo se enmarca en las actividades preliminares relacionadas con el proyecto de investigación y desarrollo “I-QuAGI: Un enfoque inteligente para la evaluación de calidad de procesos de software ágiles” (PID SIPPBRE0008092), correspondiente al Centro de Investigación Aplicada a TIC (CInApTIC) de la Universidad Tecnológica Nacional, Facultad Regional Resistencia, de la provincia del Chaco, Argentina.

## Referencias

1. Xie, T. (2018, September). Intelligent software engineering: Synergy between AI and software engineering. In *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications* (pp. 3-7). Springer, Cham.
2. Deugo, D., Oppacher, F., Kuester, J., & von Otte, I. (1999). Patterns as a Means for Intelligent Software Engineering. In *IC-AI* (Vol. 99, pp. 605-611).
3. Jurado-Muñoz, J. L., & Pardo-Calvache, C. J. (2013). La gestión de proyectos Software, una prospectiva en la aplicación de estrategias en la Ingeniería colaborativa. *Lámpsakos*, (9), 26-33..
4. PINTO, Noelia, ACUÑA, César et al. Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework. XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 47 (CABA, 2018). 2018.
5. J. Y. González, C. P. Calvache y O. S. Gómez, “Estado del Arte de la Utilización de Metodologías Ágiles y Otros Modelos en Pymes de Software”, *Informática-XVI Convención y Feria Internacional*, 2016
6. PINTO, Noelia Soledad. Framework para la evaluación de calidad de proyectos ágiles de software. 2020. Tesis Doctoral. Universidad Nacional de La Plata.
7. M. Adnan y M. Afzal, “Ontology based multiagent effort estimation system for scrum agile method”, *IEEE Access*, vol. 5, 2017.
8. L. I. U. Xu, “User Story based Information Visualization Type Recommendation System”, *International Journal of Information Engineering & Electronic Business*, 2019.
9. S. Dhir, D. Kumar y V. B. Singh, “Feedforward and Feedbackward Approach-Based Estimation Model for Agile Software Development. In *Advances in Computer and Computational Sciences*”, Springer, pp. 73-80, 2017.
10. S. Bilgaiyan, P. K. Panigrahi, y S. Mishra, “Chaos-Based Modified Morphological Genetic Algorithm for Effort Estimation in Agile Software Development. In *A Journey Towards Bio-inspired Techniques in Software Engineering*”, Springer, pp. 89-102, 2020.

11. M. Hamid, F. Zeshan, A. Ahmad, F. Ahmad, M. A. Hamza, Z. A. Khan y H. Aljuaaid, "An Intelligent Recommender and Decision Support System (IRDSS) for Effective Management of Software Projects", IEEE, 2020.
12. V. Alizadeh, M. A. Ouali, M. Kessentini y M. Chater, "RefBot: Intelligent Software Refactoring Bot", 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)", 2019
13. K. Athiththan, S. Rovinsan, S. Sathveegan, , N.Gunasekaran, K. S. A. W. Gunawardena y D. Kasthurirathna, "An Ontology-based Approach to Automate the Software Development Process", IEEE International Conference on Information and Automation for Sustainability (ICIAfS)., 2018
14. M. Murtazina y T. Avdeenko, "The ontology-driven approach to intelligent support of requirements engineering in agile software development", International Conference on Information Technology and Nanotechnology (ITNT), (pp. 1-6). May. 2020.
15. Kransdorff, Arnold. "Corporate amnesia: Keeping know-how in the company". Elsevier, 1998.
16. Chella, Antonio, et al. "Agile passi: An agile process for designing agents." International Journal of Computer Systems Science & Engineering 21.2 (2006): 133-144.
17. Cossentino, Massimo, and Colin Potts. "A CASE tool supported methodology for the design of multi-agent systems." International Conference on Software Engineering Research and Practice (SERP'02). 2002.
18. Ivar Jacobson, El Proceso Unificado de Desarrollo de Software, Addison Wesley, 2000.

## Emociones en el uso de software: Una experiencia de relevamiento utilizando PrEmo

Gabriela Tomaselli<sup>1</sup>, Nicolas Alegre<sup>1</sup>, Rodrigo Cuevas<sup>1</sup>, Cesar J. Acuña<sup>1</sup>, Noelia Pinto<sup>1</sup>

<sup>1</sup> CInApTIC – UTN FRRe, Facultad Regional Resistencia – Universidad Tecnológica Nacional  
French 414, Resistencia, Chaco, Argentina  
{gabriela.tomaselli,cesar.acuna,noelia.pinto}@ca.frre.utn.edu.ar

**Resumen.** Los constantes avances con respecto al estudio de las emociones en la ingeniería de software hacen que cada vez sean más consideradas al momento de desarrollar software. Durante los últimos años, las emociones han pasado a desempeñar un papel muy importante introduciendo conceptos como la experiencia de usuario al utilizar un determinado software. Analizar las emociones y comportamiento de las personas al usar aplicaciones de software, no se limita a detectar, medir y analizar dichas emociones o respuestas de comportamiento, sino que tiene como fin último proponer modificaciones al software que permitan mejorar su calidad en la experiencia de uso.

Con el objetivo de conocer la relación que existe entre las emociones que percibe una persona al utilizar un software y la calidad percibida del mismo, en este trabajo se presentan los resultados obtenidos al analizar las emociones percibidas durante la utilización de aplicaciones específicas.

La realización de este trabajo ha permitido comprobar que todo usuario de un determinado software percibe en mayor o menor medida diferentes emociones, permitiendo abordar como trabajo futuro investigar la medida en que estas influyen positiva o negativamente sobre la calidad percibida de las aplicaciones

**Palabras clave:** Calidad del Software, Experiencia de usuario, Emociones, Ingeniería de Software Empírica.

### 1 Introducción

Cada vez es más frecuente la introducción de conceptos como experiencia de usuario (UX) y el estudio de las emociones al utilizar un sistema en el ciclo de vida del mismo. Según Donald Norman, los objetos que nos resultan atractivos funcionan mucho mejor [1], y los sistemas de información no quedan fuera de esta afirmación, es así que en la actualidad la percepción de los usuarios está siendo considerada mucho más, fortaleciendo su relación con la calidad del producto de software. Esta percepción que se tiene de un sistema interactivo y de información está influenciada cada vez por más

aspectos, tanto internos como externos al mismo, su uso, contexto y del estado anímico. Estas propiedades son determinantes en la experiencia o sensación del usuario. Conocer el contexto y a los usuarios que harán uso del producto digital nos dará la posibilidad de delimitar nuestro campo de diseño y desarrollo, optimizando esfuerzos y recursos, para obtener un producto de calidad asegurado mejorando la relación usuario-producto [2].

Los múltiples modelos de calidad de la Organización Internacional de Estandarización (ISO) tienen en cuenta características como la funcionalidad, la eficiencia, la compatibilidad, fiabilidad, seguridad, facilidad de mantenimiento, portabilidad, calidad de interacción y la usabilidad, siendo esta última la utilizada para poder determinar la facilidad de manejo que ofrece un sistema a través de su interfaz. La ISO divide a la usabilidad en dos puntos de vista, la usabilidad como proceso, y la usabilidad como producto, teniendo como factores en común a la percepción, las emociones y la búsqueda de la satisfacción del usuario [3].

Existe en la literatura gran diversidad de definiciones para el concepto “emoción”, por lo que a efectos de esta investigación se adopta la propuesta por Ekman: “la emoción es una reacción a eventos considerados relevantes a las necesidades, metas o preocupaciones de un individuo, que existe durante un tiempo determinado (segundos, como máximo minutos)” [4].

El trabajo que aquí se presenta, se encuadra en las líneas de investigación y desarrollo del proyecto “Evaluación del impacto de las emociones en la calidad de software desde el punto de vista del usuario”; el cual, a su vez, complementa esfuerzos y resultados de investigación en proyectos de investigación previos, principalmente del proyecto predecesor denominado “Un Modelo de Evaluación de la Calidad de Procesos Ágiles de Desarrollo de Software” (2017-2019), cuyos principales resultados de investigación componen un marco de trabajo denominado AQF (Agile Quality Framework) [5] que incluye, por un lado, un modelo de calidad de software para procesos ágiles de desarrollo, denominado QuAM (Quality Agile Model) [6, 7] y, por otro lado, implica el desarrollo de una plataforma tecnológica que, a través de una herramienta de software denominada QuAGI [8] permite gestionar los elementos del modelo de calidad y evaluar fácilmente la calidad del proceso ágil asociado.

Dicho framework permite evaluar a través de un modelo de calidad específico y una herramienta que lo gestiona, la calidad de los procesos ágiles. Sin embargo, la relación que existe entre factores como experiencia de usuario y el impacto de las emociones en la calidad percibida por parte del usuario al interactuar con el software es un aspecto relacionado con el aseguramiento de la calidad que aún debe ser tratado; en consecuencia, el desafío del proyecto en el que se enmarca este trabajo será el diseño de un proceso de evaluación de calidad del software que contemple las emociones y la interacción del usuario con el producto final. Para ello se trabajará en el desarrollo de un modelo y herramientas que permitan la automatización al evaluar la calidad de software según el impacto de las emociones de usuarios finales y ejecutar acciones que permitan mejorar la calidad del proceso de desarrollo y del producto de software [9].

Como conclusión de trabajos preliminares del mencionado proyecto [10, 11] y entre otras afirmaciones, podemos decir que analizar las emociones y comportamiento de las personas al usar aplicaciones de software, no se limita a detectar, medir y analizar dichas emociones o respuestas de comportamiento, sino que tiene como fin último proponer modificaciones al software que permitan mejorar su calidad en la experiencia de uso. Incluso, en este mismo escenario, creemos que es interesante considerar la medida en que la evaluación de las emociones que experimentan los usuarios al utilizar software puede influir en el proceso de desarrollo del mismo, retroalimentando actividades o incluso generando nuevas.

Así pues, en una primera etapa de la investigación y haciendo foco en las emociones más frecuentemente percibidas por los usuarios al utilizar software, empleamos la herramienta PrEmo, desarrollada por Pieter Desmet, la cual sirve para detectar 14 emociones producidas en la interacción con un producto. Las emociones se dividen en dos grupos, las positivas en donde podemos encontrar la alegría, admiración, orgullo, confianza, satisfacción, fascinación y deseo, por otro lado, las negativas como por ejemplo la tristeza, el miedo, vergüenza, desprecio, la furia, el aburrimiento y el disgusto, cada emoción está representada mediante animaciones con expresiones dinámicas faciales y corporales, lo que convierte a PrEmo en una herramienta muy visual [12].

Se presentan en este trabajo los resultados del análisis realizado a diferentes personas usuarias de tres aplicaciones web, a través de una encuesta online como instrumento de recolección de evidencia empírica. El resto de este artículo se estructura de la siguiente manera: en la sección 2 se resume el desarrollo del Test, en la sección 3 se discuten los resultados obtenidos, expuestos a través de gráficos representativos. Finalmente, en la sección 4 se incluyen conclusiones respecto al tema abordado y se describen trabajos futuros relacionados.

## **2 Desarrollo**

En los últimos años se presentó un crecimiento en las investigaciones relacionadas al reconocimiento de las emociones en experiencias de uso de software. Aunque la literatura muestra algunos avances a nivel de los modelos teóricos, el número de implementaciones de tales procesos es escaso. Por lo que, con el objetivo de obtener un relevamiento del impacto de las emociones sobre la percepción de la calidad del software por parte de los usuarios, se recolectó evidencia empírica referente a las personas que usualmente utilizan ciertas aplicaciones web.

En trabajos anteriores, se llevó a cabo, por un lado, la revisión de las tecnologías existentes para evaluar las emociones de las personas al usar software [11], lo cual demostró que si bien existen iniciativas que utilizan la tecnología para evaluar emociones en usuarios de software, el conocimiento existente no aporta propuestas que permitan la definición de modelos o estrategias que evalúen el impacto de las en la calidad de software percibida por el usuario. Por otro lado, se realizó una revisión para conocer el impacto de

las emociones sobre la percepción de la calidad de software por parte de las empresas de la Industria del Software [13], en la que se concluyó que, en tanto solo un pequeño porcentaje de ellas ha llevado a cabo evaluación de emociones, ninguna de las que no lo hizo lo calificó como no relevante; y ante la consulta acerca de si considerasen implementar tal evaluación, la totalidad de las respuestas fueron afirmativas.

En consonancia con los trabajos realizados previamente, para el relevamiento sobre las emociones percibidas al utilizar diferentes aplicaciones se llevó a cabo una encuesta<sup>1</sup>, la cual fue dirigida a cualquier persona con acceso a sitios web. El objetivo de este relevamiento es tomar conocimiento de las emociones que perciben las personas al utilizar un determinado software para luego identificar la relación existente con la calidad del mismo.

El proceso de relevamiento fue definido en función a 3 fases: Preparación, Ejecución y Documentación, las cuales se describen a continuación.

## 2.1 Fase 1: Preparación

Para el relevamiento de la información se realizó una encuesta donde cada persona debía completar un proceso dentro de tres aplicaciones populares distintas y seleccionar, mediante una escala de tres niveles, qué tanto ha identificado una emoción durante el proceso. Esta forma de relevar información permitió obtener todos los datos necesarios para el estudio de una forma muy simple y accesible para las personas, ya que podían realizarlo desde cualquier dispositivo y en un tiempo mínimo.

La encuesta realizada se estructuró teniendo en cuenta las siguientes secciones:

- *Datos demográficos*: a fin de poder estudiar y obtener conclusiones sobre las emociones respecto a ciertas características identificadas; por ejemplo, la emoción más reconocible en un rango de edad determinado al utilizar determinadas aplicaciones web. La información requerida a la persona encuestada fue: edad, sexo, provincia/estado, país y profesión/ocupación.
- *Dispositivo desde donde se realiza el test*: con el objetivo de poder identificar qué relación tienen las emociones con el dispositivo desde donde se realice el proceso en las aplicaciones.
- *Descripción del proceso en la aplicación*: aquí se describe un proceso a realizar en una determinada aplicación que es utilizada habitualmente como, por ejemplo, agregar un producto a favoritos en un e-commerce, buscar un hotel, etc.
- *Evaluación de las emociones*: con el objetivo de identificar las emociones tanto positivas como negativas que ha experimentado la persona encuestada durante el proceso realizado en la sección previa.

Cada encuestado, recibió exactamente el mismo test con un total de tres aplicaciones en las cuales debían hacer un proceso específico en cada una y seleccionar para cada

---

<sup>1</sup> Disponible en [<https://forms.gle/tccfwkk9oZ5rf3dw8>],

emoción si definitivamente le produjo, si lo hizo por momentos o si definitivamente no le produjo dicha emoción.

La encuesta no ha sido dirigida para una población en particular, ya que el objetivo de la misma era obtener la mayor cantidad de respuestas posibles para poder obtener conclusiones más precisas. De todas formas, sí fue de total importancia obtener la información personal de cada encuestado para, una vez finalizada la encuesta, poder encontrar patrones e información útil al estudiar los resultados.

Las tres aplicaciones para evaluar las emociones de los usuarios fueron seleccionadas considerando que correspondan a aplicaciones conocidas y utilizadas habitualmente, accesibles desde cualquier dispositivo y que dispongan de algún flujo simple como un proceso de búsqueda de alguno de algún producto o servicio.

## 2.2 Fase 2: Ejecución

Luego de varias iteraciones y, una vez que la encuesta se estructuró completamente se procedió a la fase de ejecución.

Para esto se enviaron las encuestas vía e-mail y publicaciones en redes sociales con un enlace direccionando a un cuestionario realizado mediante Google Forms, el cual facilita la obtención de la información y su posterior procesamiento.

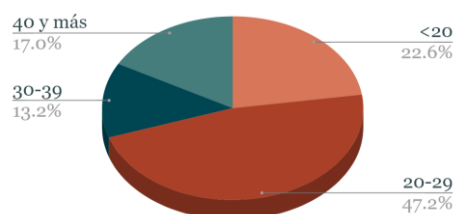
Este proceso tuvo lugar durante los meses de junio, julio y agosto de 2021, habiéndose obtenido un total de 53 respuestas; una vez finalizadas las encuestas se procedió a evaluar la información obtenida.

A continuación, la siguiente sección presenta el análisis de resultados obtenidos luego de la ejecución de la encuesta, y que constituye la Fase 3: Documentación.

## 3 Discusiones

A partir de la ejecución de las encuestas, se generó una planilla de Google con las respuestas obtenidas, a fin de tabular resultados y realizar un análisis gráfico de los mismos con el objetivo de obtener conclusiones al respecto.

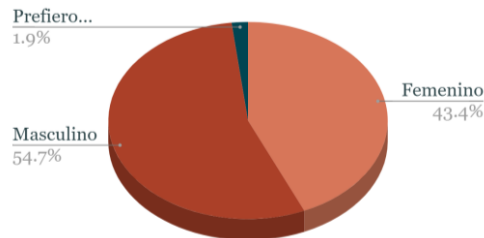
En consecuencia, al analizar los resultados obtenidos respecto a Datos demográficos, según se observa en la Figura 1 casi el 70% de las personas encuestadas son menores de 30 años, siendo la franja etaria 20-29 la que proveyó aproximadamente la mitad de las respuestas obtenidas.



**Fig. 1.** Datos Demográficos - Edad

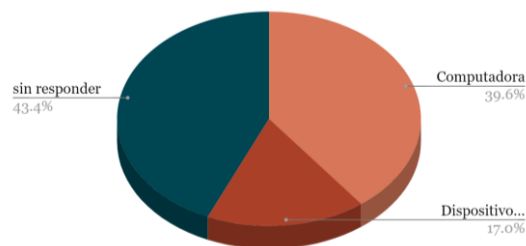


En cuanto a la división en relación al Sexo de las personas encuestadas, es prácticamente equivalente, con un ligero porcentaje mayor Masculino, según se muestra en la Figura 2.



**Fig. 2.** Datos Demográficos - Sexo

Para completar el análisis del contexto en el que se realiza la prueba, y obtener una mejor caracterización de los resultados obtenidos, se consultó también acerca del dispositivo desde donde se completó la encuesta. Las respuestas se grafican en la Figura 3, donde se destaca que alrededor del 40% lo hizo desde una Computadora (notebook, escritorio, etc.), un 17% utilizó un Dispositivo móvil (teléfono celular, tablet), en tanto que el mayor porcentaje no respondió la pregunta.



**Fig. 3.** Dispositivo desde donde se realiza el test

Luego de este panorama general, la encuesta abordó en forma más específica lo referido a las emociones percibidas por la persona encuestada durante el proceso propuesto sobre cada una de las aplicaciones consideradas.

En primer lugar, se solicitó acceder a Mercado Libre, realizar una búsqueda de producto y agregarlo a Favoritos; luego, el test requiere se identifiquen las emociones generadas, positivas y negativas, las cuales se exponen en la Figura 4.

Respecto de las emociones positivas, la más frecuentemente identificada es el deseo, casi por un 50% de las personas que realizaron el test. La satisfacción y la confianza son las emociones que siguen en percepción, mientras que en el otro extremo se ubica el orgullo, que no alcanza al 15% aun considerando los que definitivamente u ocasionalmente la percibieron.

En lo referido a emociones negativas, la gran mayoría no reporta este tipo de emociones, siendo las más referidas el aburrimiento y el disgusto, y las menos percibidas la vergüenza y la furia.

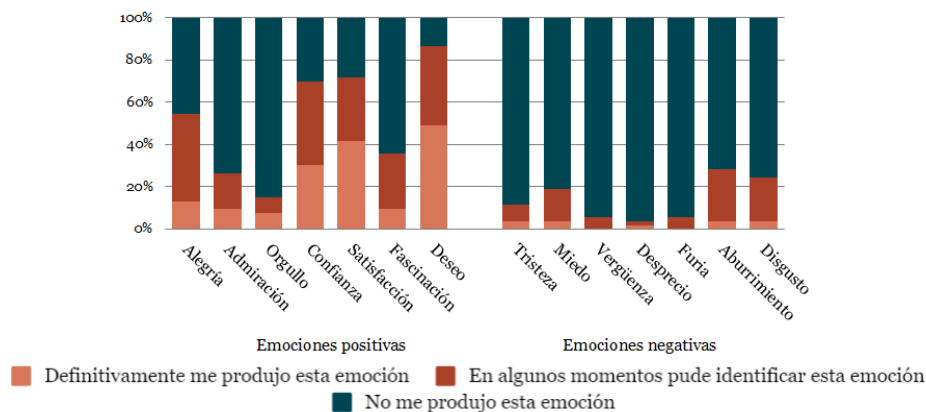


Fig. 4. Primera aplicación: Mercado Libre

En segundo lugar, se requirió acceder a Booking.com, buscar un hotel para 2 personas para la primera semana del mes siguiente y agregarlo a Favoritos; las emociones identificadas, positivas y negativas, se muestran en la Figura 5.

Al igual que en el caso anterior, el deseo es la emoción positiva más percibida, casi por un 90% entre quienes la percibieron; pero a diferencia de Mercado Libre, surge la alegría como segunda emoción entre quienes la reportan, en tanto hay coincidencia en la emoción menos percibida, el orgullo.

En cuanto a emociones negativas, nuevamente no son percibidas por la mayoría, aunque se presentan porcentajes mayores que en el caso de la aplicación anterior, con más de un 25% reportando disgusto, y porcentajes similares en tristeza y aburrimiento.

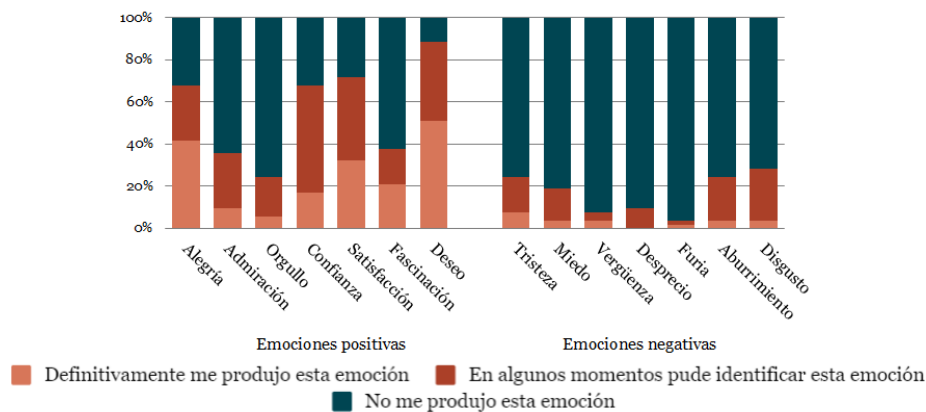


Fig. 5. Segunda aplicación: Booking

Por último, se pidió acceder a Flybondi, buscar algún vuelo desde Corrientes a Buenos Aires para un pasajero la primera semana del mes siguiente, y repetir el proceso de identificación de emociones, cuyos resultados pueden observarse en la Figura 6.

Al igual que en los casos anteriores, el deseo resulta la emoción positiva más percibida, aunque en un porcentaje menor, de aproximadamente el 75%, alcanzando en esta aplicación un porcentaje similar la satisfacción.

En cuanto a emociones negativas, si bien son percibidas en mucha menor medida que las positivas, los porcentajes son mayores que en las otras dos aplicaciones, superando el 25% el disgusto, el aburrimiento y el miedo; al igual que en la evaluación de Mercado Libre, la vergüenza no fue percibida en ningún caso.

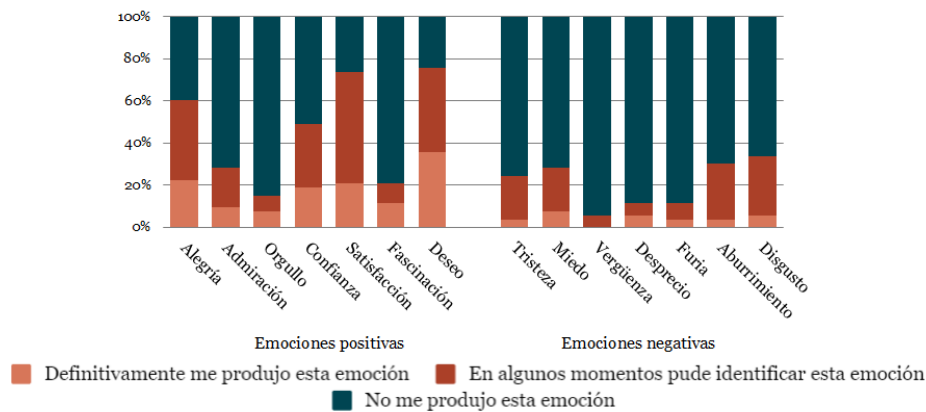


Fig. 6. Tercera aplicación: Flybondi

A los efectos de efectuar una comparación final y extraer conclusiones, la Figura 7 muestra un resumen de las emociones -tanto positivas como negativas- reportadas en el test realizado.

Aquí es posible observar que deseo, satisfacción y confianza resultan las emociones positivas más percibidas, lo que podría relacionarse con una mejor valoración del software evaluado, en tanto el disgusto y el aburrimiento llevan la delantera en cuanto a emociones negativas, hecho éste que podría impactar negativamente en la evaluación de calidad del software usado.

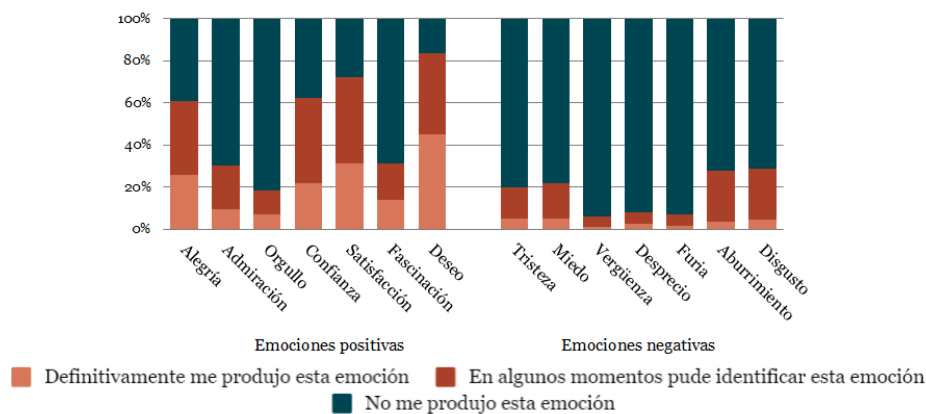


Fig. 7. Totales

## 4 Conclusiones y Trabajos Futuros

El análisis de las emociones durante la utilización de un determinado software se convierte en un factor cada vez más importante. Por tanto, en este trabajo se presenta un relevamiento del impacto que poseen dichas emociones sobre la percepción de la calidad del software. Es decir, la relación existente entre las emociones que percibe una persona al utilizar un software y la calidad percibida del mismo.

Los resultados de esta investigación ponen de manifiesto la diversidad de emociones que se producen en las interacciones de personas con software y, en consecuencia, la importancia que se debe dar a dichas emociones al estudiar la relación que poseen con el impacto sobre la percepción de la calidad del software. Asimismo, el estudio permite definir la hipótesis que mientras mayores sean las emociones positivas percibidas por una persona al utilizar un software, mayor será la tendencia a percibir una calidad más aceptable del producto.

La realización de este trabajo ha permitido comprobar que todo usuario de un determinado software percibe en mayor o menor medida diferentes emociones y puede resultar en un factor determinante al momento de evaluar la calidad de este software.

Como trabajos futuros, se pretende, a partir de los resultados presentados en este artículo, y haciendo uso de la técnica de “Focus Group”, llevar adelante un proceso de investigación cuali-cuantitativa que se centre en la observación de ciertas emociones en experiencias de uso de software, bajo un contexto controlado del que participarán usuarios y evaluadores. Esto contribuirá, asimismo, con la definición más precisa de un modelo de evaluación de calidad centrado en la relación entre emociones generadas desde la experiencia de uso y la percepción de calidad del software por parte del usuario.

## 5 Agradecimientos

Este trabajo se enmarca en las actividades relacionadas con el proyecto de investigación y desarrollo “Evaluación del impacto de las emociones en la calidad de software desde el punto de vista del usuario” (PID: SIUTIRE0005517TC), del Centro de Investigación Aplicada en Tecnologías de la Información y Comunicación (CInApTIC), de la Facultad Regional Resistencia de la Universidad Tecnológica Nacional.

## Referencias

- [1] Norman, D.A. (2004). El diseño emocional: por qué nos gustan o no los objetos cotidianos. Ed. Paidós.
- [2] Hotimsky, T., Molina, W., Salgado, C., Peralta, M., Sánchez, A. (2019). “Cambiano Percepciones: UX Designs”. *Proceedings del WICC 2019*, Universidad Nacional de San Juan).

- [3] González-Sánchez, J.L., Montero-Simarro, F., & Gutiérrez-Vela, F.L. (2012). Evolución del concepto de usabilidad como indicador de calidad del software. *Profesional de la Información*, 21(5), 529–536.
- [4] Ekman, P. (1994). Moods, emotions, and traits. *The nature of emotion: Fundamental questions*, 56-58. Oxford University Press.
- [5] Pinto, N., et.al. (2018). Quality evaluation of agile processes: Measurement of requirements management using AQF v2. *Proceedings QUATIC 2018, Coimbra, Portugal*, p. 15-20.
- [6] Pinto, N., Acuña, C.J., & Cuenca Pletsch, L.R. (2016). Quality Evaluation in Agile Process: A First Approach. *Proceedings CACIC 2016, San Luis, Argentina*, p. 525-534.
- [7] Cuenca Pletsch, L.R., Acuña, C. J., Tomaselli, G., & Pinto, N. (2017). QUAGI: Una propuesta para el seguimiento y evaluación de proyectos de Software Ágiles. *Anais do V SABTIC, VIII STIN e XVIII Fórum de Informática, Três de Maio, Brasil*, p. 127-136.
- [8] Pinto, N., et.al (2018). Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework. *Proceedings ASSE-JAIIO 47, CABA, Argentina*, p. 61-72.
- [9] Acuña, C.J., et.al. (2020). Evaluación del impacto de las emociones en la calidad de software desde el punto de vista del usuario. *Proceedings del XXII Workshop de Investigadores en Ciencias de la Computación (WICC 2020), El Calafate, Santa Cruz, Argentina*, p. 381-385.
- [10] Pinto, N., Tomaselli, G., Torres, D., & Acuña, C.J. (2020). Hacia la evaluación de emociones en experiencias de uso de software: Una revisión sistemática. *Proceedings de CONAISI 2020, Universidad Tecnológica Nacional – Facultad Regional San Francisco*, p. 74-81.
- [11] Tortosa, N., et.al. Evaluación del impacto de las emociones en la Calidad del Software: Una revisión sistemática de tecnologías para evaluar emociones de personas al usar software. (2020). *Proceedings del 2020 IEEE Congreso Bienal de Argentina (ARGENCON)*.
- [12] Pérez, G. (2013). Diseño Emocional: Metodologías y herramientas para cuantificar emociones - Casiopea. Recuperado el 21 de septiembre de 2021, de [https://wiki.ead.pucv.cl/Diseño Emocional: Metodologías y herramientas para cuantificar emociones](https://wiki.ead.pucv.cl/Diseño_Emocional:_Metodologías_y_herramientas_para_cuantificar_emociones)
- [13] Tomaselli, G., Acuña, C.J., Pinto, N., & Torres, D. (2021, octubre). Vinculación Universidad-Industria: Relevamiento sobre Impacto de las Emociones en Calidad de Software. *Proceedings del 5° Congreso Argentino de Ingeniería (CADI), Buenos Aires, Argentina*.

## Desarrollo de entornos educativos interactivos digitales mediante Design Thinking

Alejandro Micheloud<sup>1</sup>, Eduardo Zamudio<sup>1</sup>

<sup>1</sup> Instituto de Investigación Desarrollo e Innovación en Informática  
IIDII, FCEQyN, UNaM, Argentina  
alejandromicheloud@gmail.com, eduardozamudio@fceqyn.unam.edu.ar

**Resumen.** Los entornos digitales interactivos en educación buscan mejorar el aprendizaje mediante el uso de la tecnología. Sin embargo, el desarrollo de soluciones tecnológicas para este propósito presentan varios desafíos en etapas tempranas del desarrollo, en particular el diseño. Design Thinking proporciona una manera de abordar problemáticas en distintas disciplinas, con foco en las personas. En los últimos años se ha adaptado a diversos ámbitos siendo utilizada en Google, IBM, SAP y otros grandes de la industria.

El presente trabajo propone una adaptación de Design Thinking en el desarrollo ágil de productos software para el desarrollo de entornos educativos interactivos digitales. La adaptación abarca la definición de fases, procesos, roles, entradas y salidas. La adaptación de Design Thinking propuesta se valida en un caso de estudio real en el desarrollo de una plataforma educativa interactiva digital.

**Palabras clave:** Design Thinking, Aprendizaje mejorado por tecnología, Producto Software, Desarrollo Ágil.

### 1 Introducción

En las últimas décadas se han incorporado herramientas tecnológicas en los procesos educativos. Tanto dentro como fuera de las instituciones educativas los protagonistas de la Educación Superior utilizan tecnología de manera habitual.

Technology-Enhanced Learning (TEL) es un campo que estudia el aprendizaje mejorado por tecnología como un tema relevante, cuya importancia es reconocida por investigadores educativos, profesionales, diseñadores de software y responsables políticos. En el aprendizaje mejorado con tecnología se interceptan diferentes disciplinas, como la informática, la educación, la psicología, la filosofía, la pedagogía y la comunicación [1].

Uno de los problemas importantes es identificar en etapas tempranas el objetivo general del producto software a desarrollar. Al momento de desarrollar una plataforma educativa interactiva digital, se deben considerar los distintos roles ocupados por las personas, cada una de ellas normalmente tendrá un punto de vista diferente. Por este motivo, es necesario identificar los aspectos importantes intrínsecos a la actividad que se llevará a cabo, asignar prioridades a las necesidades y establecer acuerdos entre las partes [2].

Design Thinking se destaca por el diseño de productos centrados en las personas, ya que es una propuesta para resolver problemas complejos [3].

El objetivo de este trabajo es relevar y describir el Design Thinking y los marcos de trabajo relevantes para adaptarlo a la producción de software en entornos educativos interactivos aplicando principios de desarrollo ágil.

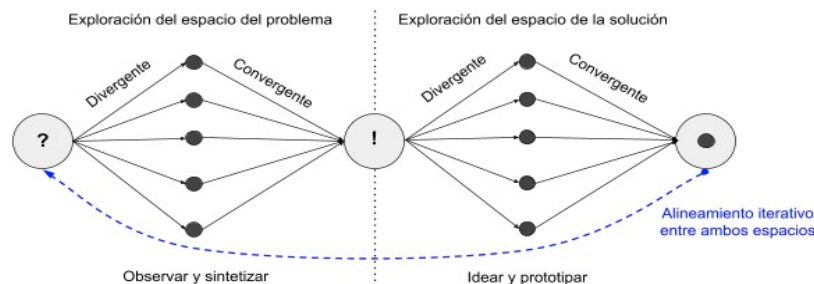
Este documento se organiza de la siguiente manera. La sección 2 describe los Materiales y métodos utilizados para el desarrollo de la propuesta. La sección 3 presenta los trabajos relacionados con la problemática del desarrollo de entornos educativos interactivos digitales. La sección 4 presenta una propuesta de adaptación de Design Thinking en la problemática. La sección 5 presenta un caso de estudio real donde se aplica la propuesta junto con sus resultados. La sección 6 presenta las conclusiones generales del trabajo.

## 2 Materiales y Métodos

Design Thinking es una herramienta pensada para el diseño de soluciones con foco en la innovación, considerando la amplitud de necesidades en forma global. Es incremental, altamente interactivo, e impulsado por personas con diferentes antecedentes y experiencias [4]. La innovación guiada por Design Thinking complementa la visión tradicional del diseño de software, aportando innovación a productos y servicios [5].

El diseño es una actividad que trata con los significados. Design Thinking permite generar soluciones con nuevos significados, que activan diversos elementos cognitivos, emocionales y sensoriales, involucrados en la experiencia humana.

Design Thinking trata con dos espacios fundamentales. Por un lado, el espacio de problemas y soluciones, y por el otro, el pensamiento divergente-convergente [6]. El espacio del problema trata de la adquisición de una comprensión intuitiva de la problemática.



**Fig. 1.** Espacio del problema y espacio de solución, ciclo iterativo en Design Thinking basado en (Lindberg, Meinel, & Wagner, 2011). [Elaboración propia].

Principalmente se concentra en observar casos de uso generales o escenarios representativos planteados por las personas intervinientes. El espacio de solución promueve la evaluación de un gran número de ideas alternativas en paralelo, mediante bocetos y prototipos de solución. De esta manera, se busca establecer un primer paso para transformar las ideas en representaciones. La **Figura 1** presenta los conceptos de espacio de problema y espacio de solución desde la perspectiva de Design Thinking, donde se destaca el pensamiento convergente-divergente desde una perspectiva iterativa, basado en [7].

La alineación iterativa de ambos espacios representa las ideas y conceptos que facilitan la comunicación, no solo en el equipo de diseño, sino también con usuarios, clientes y expertos. Design Thinking ayuda a mantenerse en contacto con el entorno relevante para el problema y puede usar esta información para refinar y revisar alternativas de solución [8].

El Design Thinking produce un sistema de frenos y contrapesos para contribuir a soluciones equilibradas, innovadoras y adecuadas. Estas características son necesarias en entornos desafiantes como los entornos educativos interactivos digitales [8].

### 3 Trabajos relacionados

Los marcos de trabajo son instancias basadas en Design Thinking definidos para estandarizar las etapas y procesos. La Tabla 1 presenta definiciones de marcos de trabajo destacados asociados a diversas organizaciones.

**Tabla 1.** Caracterización entre marcos de trabajo.

Marco de trabajo	Etapas	Descripción
Hasso Plattner Institute of Design at Stanford [7][9]	1. Empatizar 2. Definir 3. Idear 4. Prototipar 5. Testing	Ideado para la educación primaria y secundaria (sector K12), para ser aplicada por docentes y estudiantes en la resolución de problemas de índole general.
IDEO [10]	1. Descubrir 2. Interpretar 3. Idear 4. Experimentar 5. Evolucionar	Centrado en las necesidades de las personas, las posibilidades tecnológicas y los negocios.
Google Design Sprints [11]	1. Comprender 2. Definir 3. Bosquejar 4. Decidir 5. Prototipar 6. Validar	Aplicado al diseño y desarrollo de producto, optimizada para startups y emprendedores.
Deep Design Thinking [12]	1. Descubrir 2. Empatizar 3. Experimentar 4. Producir	Aplicado en el contexto del aula como herramienta pedagógica dentro del sector K12.
SAP design [13]	1. Descubrir: alcance, investigación, síntesis. 2. Diseño: idear, prototipar, validar. 3. Entregar: implementar, testear, desplegar.	Utilizado como enfoque de innovación en la relación con clientes.
Double diamond, Design Council UK [14]	1. Descubrir: identificar, investigar y comprender el problema inicial. 2. Definir: delimitar y definir un problema a resolver. 3. Desarrollar: centrarse y desarrollar una solución. 4. Entregar: probar y evaluar, producir y lanzar.	Proporciona una representación gráfica del proceso de diseño.
Design for America [15]	1. Identificar 2. Sumergirse 3. Replantear 4. Idear	Promovido por universidades con campus de estudios distribuidos en Estados Unidos de América.



	5. Construir	
	6. Testear	
Austin Center for Design [16]	1. Identificar	Enfocado en la resolución de problemas sociales a través del diseño y la educación en diseño.
	2. Descubrir	
	3. Idear	
	4. Iterar	
	5. Refinar	
	6. Validar	
	7. Implementar	

Los marcos de trabajo propuestos en Design Thinking tienen etapas en común. Algunas etapas se comparten entre todas las propuestas.

Los marcos de trabajo proponen un desarrollo ágil e iterativo cuyas etapas se pueden generalizar en **empatizar**, incluyendo las tareas de comprender, identificar, descubrir; **idear**, incluyendo las tareas de construir, experimentar, diseñar, prototipar, producir; y **testear**, incluyendo las tareas de implementar y evolucionar.

La generalización de etapas y la reducción en la cantidad de las mismas podría contribuir en la adopción de la metodología.

#### 4 Adaptación de Design Thinking al desarrollo entornos educativos interactivos digitales

En la producción de software, Design Thinking puede representar un eslabón importante en el desarrollo de productos. Su objetivo es crear una especificación de producto, que se representa mediante un conjunto de artefactos. Esos artefactos pueden incluir Product Requirement Documents (PRD), prototipos, roadmaps, épicas e historias de usuario del producto software que se pretende construir.

La especificación del producto es importante tanto en productos nuevos, que aún no han sido desarrollados, como así también en productos existentes.



Fig. 2. Design thinking integrado en el desarrollo de producto software. [Elaboración propia].

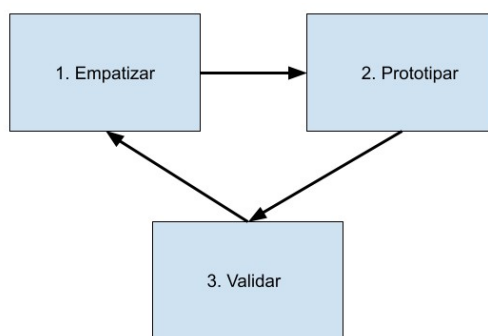
En productos nuevos, la aplicación de Design Thinking tiene como objetivo especificar un Producto Mínimo Viable (MVP). Mientras que en productos existentes, su objetivo es incorporar nuevas funcionalidades y/o experiencias de usuario.

En la **Figura 2** se integra el proceso de Design Thinking dentro de los diferentes eslabones que componen la producción de software.

#### 4.1 Proceso de Design Thinking

El proceso de Design Thinking está compuesto por las etapas empatizar, prototipar y validar. Se puede realizar de manera iterativa cuando el producto software a desarrollar es extenso, es decir, intervienen varios roles de usuario o bien, se pretende desarrollar un gran conjunto de funcionalidades.

Realizarlo de manera iterativa posibilita a su vez enfocarse en roles de usuarios o funcionalidades específicas. Por ejemplo, se pueden organizar procesos de Design Thinking para el rol docente, alumno, o para la funcionalidad proyectar en clase, o la funcionalidad de evaluaciones.



**Fig. 3.** Etapas del proceso Design Thinking. [Elaboración propia].

En cada una de las etapas del proceso Design Thinking se deben tener en cuenta los objetivos que se persiguen. Por ejemplo, desarrollar una plataforma, que permita la interacción en el ámbito educativo mediante la utilización de dispositivos móviles.

En relación al objetivo de este trabajo, basado en la propuesta de adaptación de Design Thinking para el desarrollo de entornos digitales interactivos en educación superior, es posible definir requisitos iniciales, como por ejemplo:

- La plataforma educativa deberá ser accesible desde dispositivos con Internet de uso habitual por los usuarios del sistema, como profesores y alumnos. Los dispositivos pueden incluir desktops, notebooks, tablets y/o smartphones. Las interfaces de usuario se deberán ajustar a distintas resoluciones de pantalla. La tecnología utilizada en las aplicaciones cliente deberá ejecutarse en todos los dispositivos mencionados considerando las características de conectividad de los mismos.
- Los usuarios tendrán experiencias interactivas, los docentes podrán definir materiales de estudio, tareas y entregas de tareas.
- El enfoque interactivo debe permitir a los docentes proponer dinámicas de trabajo que requieran interacción social y ayuda mutua entre los estudiantes.

El proceso Design Thinking se vincula con principios ágiles mediante la definición del MVP, ya que la misma permite identificar un conjunto de artefactos de utilidad para comenzar con el desarrollo del producto en un escenario de menor incertidumbre.

## 4.2 Artefactos

Los artefactos generados en la aplicación de Design Thinking son un PRD, prototipo, roadmap, épicas e historias de usuario [17].

**Tabla 2.** Artefactos creados en la aplicación de Design Thinking.

Artefacto	Descripción
PRD	Un documento de requerimientos de producto. Está escrito de manera tal que las personas puedan comprender lo que debe hacer un producto.
Prototipo	Los prototipos definen las interfaces de usuarios y funcionalidades del producto de software. Pueden ser de baja o alta fidelidad. Si fueran de baja fidelidad pueden estar diseñados directamente en papel o bocetos digitales simples. Caso contrario, pueden ser interfaces digitales navegables, en una versión demostración, pudiendo incluir las interacciones que realice el usuario con los elementos de la interfaz de usuario, representando así de manera fiel, la experiencia que tendrá el usuario final cuando utilice el producto software.
Roadmap	El Roadmap, conocido en español como hoja de ruta, funciona como un cronograma para la planificación del proyecto. Se trata de un manifiesto ágil, un documento directivo que enumera la lista de tareas a ejecutar en el marco del proyecto según un principio de iteración e integración continua.
Épicas	Las épicas son un conjunto de trabajo grande que puede dividirse en tareas específicas (denominadas historias de usuario) en función de las necesidades o solicitudes de los clientes o usuarios finales.
Historias de usuario	Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo una función de software proporcionará valor al cliente.

## 4.3 Desarrollo tecnológico ágil

El desarrollo tecnológico es la etapa de programación del producto software utilizando principios ágiles, es el segundo proceso indicado en la **Figura 2**, proceso que consume los artefactos producidos en la primera fase.

En el desarrollo de software los actores principales son el equipo tecnológico junto al equipo de calidad. En esta fase se realiza la programación e integración del producto de software propiamente dicho. Dependiendo de la naturaleza del producto podrá requerir desarrollo backend, frontend, aplicaciones nativas, aplicaciones de escritorio, bases de datos, entre otros.

El equipo tecnológico son los profesionales informáticos que codifican el software.

Dependiendo del tamaño, tiempos y recursos disponibles para integrar el producto software se podrán asignar varios recursos técnicos a un área, o un solo recurso para

varias áreas, esto dependerá de las características del producto software y la envergadura de la organización o empresa.

El equipo de control de calidad QA (por sus siglas en inglés Quality Assurance) se encarga de validar el funcionamiento de cada épica e historia de usuario que el Equipo tecnológico pasa a estado “Testing”. Se testean y validan dos tipos de tareas:

- Features, nuevas funcionalidades que no existían previamente en el producto software, normalmente vinculado a épicas e historias de usuario.
- Hotfixes, solución a problemas detectados en el producto de software existente, los problemas, normalmente llamados bugs, pueden ser encontrados en producción o en entornos de preproducción, como suelen ser entorno de desarrollo o entorno QA.

#### **4.4 Producto software**

El producto software es el objetivo final buscado. Por más que estemos hablando de software, algo intangible por naturaleza, y estemos trabajando en las primeras iteraciones de desarrollo, es deseable tener una versión entregable al final de cada sprint, algo “tangible” que pueda ser validado en una “demo”.

Las demos, son demostraciones del producto de software que se está construyendo. Se proponen como reuniones que deben programarse con anticipación, con fecha posterior a la finalización del sprint, los integrantes de la reunión asisten de manera remota a través de Internet. En la reunión de demostración se enseña el producto software, por ejemplo, a través de pantalla compartida, se realiza el seguimiento de las historias de usuario en las que se ha trabajado.

Si el producto de software abarca gran cantidad de lógica de negocio, que se traduce a muchas historias de usuario, en las demos se seleccionarán las historias de usuario más relevantes, en caso contrario, si el producto de software es acotado o un MVP, la demostración será completa, incluyendo todas las historias de usuario.

El equipo encargado de desarrollo y operaciones, o simplemente DevOps, intentará garantizar la integración continua / entrega continua. Se busca garantizar que el equipo tecnológico pueda integrar de manera continua, teniendo siempre la posibilidad de realizar entregas.

Se debe definir un conjunto de entornos, por ejemplo, entorno de desarrollo, entorno de calidad, entorno de producción. La cantidad de entornos de trabajo necesarios dependerá del producto de software, sus características y su presupuesto.

Para los casos donde exista un SLA (Por sus siglas en inglés Service Level Agreement), existirán cláusulas de contrato a cumplir en la etapa de explotación del software, cuando los usuarios finales utilizan el producto software. El SLA es un acuerdo escrito entre un proveedor de servicio y su cliente con objeto de fijar el nivel acordado para la calidad de dicho servicio.

## 5 Caso de estudio

La propuesta de adaptación de Design Thinking se implementó en un caso de estudio real en la empresa de diseño y software Oneclick S.L. [18] con sede en Madrid, España. El producto de software resultante forma ahora parte de su base de conocimiento. El objetivo del MVP se logró en menos de seis meses y antes de cumplirse el año comenzó a utilizarse el producto software en entorno producción con usuarios finales. A continuación se describe el caso de estudio con la adaptación de Design Thinking para el desarrollo de entornos educativos interactivos digitales, a través de sus etapas.

### 6.1 Etapa I - Empatizar

Para empatizar y conocer a los usuarios del producto software se desarrolló un workshop basado en la proposición de puesta en valor del producto [19]. En el workshop participaron docentes, directivos de colegios, diseñadores, editores y responsables técnicos de diversas áreas.



**Fig. 4.** Workshop de puesta en valor. [Elaboración propia].

Como resultado del workshop se obtienen las historias de usuario más relevantes requeridas para la definición del MVP, con historias de usuario y roadmap del producto software.

El workshop se desarrolló en forma presencial con participantes en forma virtual mediante la plataforma Miro [20].

### 6.2 Etapa II - Prototipar

El prototipo resultante incluyó 304 pantallas de interfaces de usuario para la solución, describiendo experiencias de los usuarios con roles de docente y estudiante para navegadores web, dispositivos móviles y aplicaciones nativas.

### 6.3 Etapa III - Validar

Para validar la definición del producto software se utilizaron las versiones más interactivas de los prototipos. La validación consistió en realizar una demostración a pantalla compartida simulando el uso del producto software por parte de los diferentes roles de usuario recorriendo las diferentes historias de usuario.

Como resultado del proceso de validación se obtuvieron comentarios y correcciones que permitieron depurar los prototipos consiguiendo de esta manera satisfacer de mejor manera las necesidades y requerimientos de los usuarios.

### 6.4 Artefactos resultantes

En esta sección se detallan los artefactos generados en el proceso Design Thinking que conforman las especificaciones de entrada para la implementación del producto software.

#### Épicas

Las épicas obtenidas por el proceso Design Thinking fueron 32. Algunas de las cuáles fueron las siguientes:

- Registro de un profesor y registro de alumno.
- Un alumno revisa qué tareas tiene pendientes de realizar.

#### Historias de usuario

Las historias de usuario producidas por el proceso Design Thinking alcanzaron las 109. Algunas de las cuáles se listan a continuación.

- Dado que soy Profesor, evalúo la entrega de un alumno.
- Dado que soy Profesor, veo las tareas que ya he corregido.

## 6 Conclusiones

Design Thinking ha demostrado versatilidad para adaptarse a diferentes disciplinas.

En el presente trabajo, la adecuación y aplicación de Design Thinking permitió establecer las bases para el desarrollo de una plataforma educativa interactiva centrada en usuarios finales.

La adecuación de Design Thinking como base de desarrollo ágil permitió la generación de artefactos ágiles como ser las épicas e historias de usuarios. La adopción de Design Thinking permitió establecer el punto de partida para el desarrollo de producto de software.

En adelante, se pretenden evaluar métricas de desempeño y de calidad que permitan determinar el impacto de la adopción de la incorporación de procesos como el de Design Thinking en el desarrollo de soluciones ágiles de software.

## Referencias

- [1] N. Balacheff, S. Ludvigsen, T. de Jong, A. Lazonder, and S. Barnes, *Technology-Enhanced Learning*. 2009.
- [2] G. Melles, "Design thinking in higher education: interdisciplinary encounters," *Springer Nat.*, 2020.
- [3] T. Brown, "Design thinking," *Harv. Bus. Rev.*, vol. 86, no. 6, 2008.
- [4] G. Gabrysiak, H. Giese, and A. Seibel, "Towards next-generation design thinking II: Virtual multi-user software prototypes," *Des. Think. Res. Stud. Co-Creation Pract.*, pp. 107–126, Jan. 2012.
- [5] K. Krippendorff, *The Semantic Turn*, vol. 4, no. 3. CRC Press, 2005.
- [6] B. Lawson, *How designers think \_resumen*. Routledge, 2005.
- [7] T. Lindberg, C. Meinel, and R. Wagner, "Design Thinking: A Fruitful Concept for IT Development?," *Des. Think.*, pp. 3–18, 2011.
- [8] S. Grönman and E. Lindfors, "The Process Models of Design Thinking," *Des. Technol. Educ.*, vol. 28 (2), no. 2016, pp. 110–118, 2021.
- [9] D. M. Murray, S. P. Varnell, and J. Blitstein, "Stanford dSchool," *American Journal of Public Health*, 2014. [Online]. Available: [dschool.stanford.edu](https://dschool.stanford.edu). [Accessed: 23-Oct-2021].
- [10] "IDEO." [Online]. Available: <https://www.ideo.com>. [Accessed: 23-Oct-2021].
- [11] L. Vinet and A. Zhedanov, "Design Sprint." [Online]. Available: <https://designsprint.org/es/>. [Accessed: 23-Oct-2021].
- [12] R. Mugerauer, "Deep Design," *Environmental Ethics*, 2000. [Online]. Available: <https://www.deepdesignthinking.com>. [Accessed: 23-Oct-2021].
- [13] "SAP's Human-Centered Approach." [Online]. Available: <https://experience.sap.com/designservices/approach>. [Accessed: 23-Oct-2021].
- [14] S. Wright, "Design Council," *Paper Europe*, 1999. [Online]. Available: <https://www.designcouncil.org.uk>. [Accessed: 23-Oct-2021].
- [15] E. Gerber, "Design for America," *Interactions*, 2014. [Online]. Available: <https://designforamerica.com>. [Accessed: 23-Oct-2021].
- [16] J. Kolko, "Austin center for design [www.ac4d.com](http://www.ac4d.com)," *Interactions*, vol. 20, no. 3, pp. 84–87, 2013.
- [17] M. S. Mirza and S. Datta, "Developing Software Using Agile and Design Thinking Framework," *Proc. - 2020 Int. Conf. Comput. Sci. Comput. Intell. CSCI 2020*, pp. 1819–1823, 2020.
- [18] "Oneclick, Technology for an A+ Education." [Online]. Available: <https://www.oneclick.es>. [Accessed: 23-Oct-2021].
- [19] "Value Proposition Canvas Template - Peter J Thomson." [Online]. Available: <https://www.peterj-thomson.com/2013/11/value-proposition-canvas/>. [Accessed: 23-Oct-2021].
- [20] T. C. Morphology, "Miro." [Online]. Available: <https://miro.com>.

## A relationship between ISO 9001:2015 and agility practices

Álvaro Ruiz de Mendarozqueta<sup>1</sup>, Fabio O. Bustos<sup>2</sup> and Pedro E. Colla<sup>3</sup>

<sup>1</sup> aruizdemendarozqueta@gmail.com, <sup>2</sup>fabio.oscar.bustos@gmail.com, <sup>3</sup> colla.pedro@uader.edu.ar  
Universidad Tecnológica Nacional – Regional Córdoba<sup>1</sup>  
Córdoba – Córdoba - Argentina

**Abstract.** This paper presents a preliminary mapping between ISO 9001:2015 requirements and the practices deployed in the context of the Agile process framework given by Scrum. The degree of ISO requirements coverage provided by the Scrum ceremonies and artifacts is evaluated. The conclusions can be used to understand the drivers that may ease the adoption of a formal certification framework like the ISO9001 by organizations working under Scrum, in order to increase their competitiveness in domestic and offshore markets. Some gaps are also identified and a preliminary strategy to address them is proposed.

**Keywords:** ISO 9001:2015. Agile, SCRUM, Software Engineering

### 1 Background

A research effort aiming to understand how organizations perceive the relationship between agile methodologies and traditional software engineering practices has been carried out by the authors as a stage of a larger research program (Ruiz de Mendarozqueta, Bustos, & Colla, Agile and software engineering - an invisible bond, 2020), as part of it an experiment using the data gathered using a survey among software organizations in Argentina, and the analysis of its results throw some additional light on the subject. In these software organizations, activities are executed for the development of standard products and customized implementations, updates, and maintenance of existing products, as well as embedded applications for electronic devices. Software size and complexity are increasing rapidly and the total software staff is growing continuously; still, most of the work is performed at *Small and Medium Enterprises* (SMEs or PyME in Spanish) organizations (OPSSI, 2016). Previously published efforts (Ruiz de Mendarozqueta, Bustos, & Colla, 2020) from the authors have been focused on building a preliminary discussion about the importance of addressing best, well-established practices from software engineering-based methodologies when adopting agile methods and premises. Also, a detailed discussion on how the agile and software engineering concepts are strongly bonded, even if this relationship is not often highlighted by the bibliography, has been made (Ruiz de Mendarozqueta, Bustos, & Colla, Agile and software engineering - an invisible bond, 2020), finally, some insights on the actual relations between the different factors in Argentina where modelled and some preliminary conclusion on the relevant dependent and independent variables at play were made showing that if strong software engineering practices aren't used the intrinsic value yield by the agility is eroded or even vanished (Ruiz de Mendarozqueta; Bustos; Colla, 2019)

The usage of agile methodologies bears relevance to this software industry segment, as well as the deployment of sound software engineering practices as in the local and regional demanding technology

---

<sup>1</sup> Work partially funded by grant PID SIUTNCO0004902



markets, as well as customers from the US and Europe that routinely ask provider organizations to present objective proof of their Software Engineering capabilities; in some cases even requiring the adherence to some formal quality model such as ISO9000 (ISO, 2020), SEI-CMMI (Program, 2010) or CoBIT (Elue, 2020) maturity levels, as a condition to compete.

Most scenarios and business results captured by the bibliography (Johnson, 1995) reflect the experiences of large-scale organizations formally adopting software engineering practices and embracing a formal approach to demonstrate that, leaving smaller ones wondering whether a formal approach is realistic for them, frequently leading to the prior estimation that formal initiatives are simply outside their realm of possibilities. Even though software engineering deployment efforts made at SMEs sized companies have already been documented, the focus is often placed on qualitative or methodological factors rather than quantitative ones.

In spite of the strong empirical evidence, backed by the bibliography, that software engineering best practices are good to embrace, unconditionally, regardless of the company's business context; it is an acknowledged fact that these practices are not embraced by many organizations among the SME landscape.

At the same time, the deployment of sound software engineering practices is known to be required to stay competitive in terms of productivity, quality, and schedule compliance, especially in the demanding offshore technology markets.

This need for organizational deployment of software engineering practices has been challenged by several authors, by whom the actual affordability and suitability of formal software engineering-oriented improvement initiatives for SMEs, is questioned from different perspectives.

Previous work from the authors (Ruiz de Mendarozqueta; Bustos; Colla, 2019) described a comprehensive framework that helps to understand the organizations attempting to implement software process improvement (SPI) initiatives and allows modeling the different organizational parameters involved in the business decision, the outcome that might be expected, and the level of risk associated with it.

Organizations at the SME size level embrace agility in their quest to achieve good technical or business results into their software development efforts, but formal adherence to some structured and certifiable quality framework is often perceived by SME's as a luxury they cannot afford and, therefore, they tend to avoid adopting them, unless specific competitive pressures move the organization in that direction. This situation creates a dilemma for organizations that are willing to compete at larger scales at international markets, or at business segments where the capabilities provided by strong software engineering practices are needed to survive. Contracts at a fixed cost or warranty provisions (Gey Fried Langer, 2015) are, among many others, typical scenarios where commercial willingness or risk-tolerant decisions are not enough. Even not created specifically for that purpose, formal quality accreditations boost the competitiveness of the organizations by providing a differentiator among competitors in the markets they choose to serve and as key to open new markets (Gey Fried Langer, 2015).

Besides competitive pressure factors being involved, management of software development organizations often feels they can deploy agile practices and grab all the benefits offered by them but avoid the long-term commitment and subsequent investment needed to be formally assessed under a specific framework or reference model.

Previous work from the authors (Ruiz de Mendarozqueta; Bustos; Colla, 2019) explores the value gained by applying agile methods and how that value can be eroded, even wiped out, if the execution excellence does not pay attention to classical software engineering factors. Formal certification is sometimes derogated as having the sole business value of a "paper in the wall", but close inspection of the success stories of organizations that transit the required investment and commitment found that the formal certification brings to the organization, not only a shiny token of achievement that might or might not help from a market

differentiation standpoint, but also the inception of the whole organization into rigorous management of key aspects of the development process. This additional focus helps to protect the value eroded by the inconsistent execution of agile practices.

The bibliography (Good, 2003) brings examples of a mapping between the usage of agile methodologies, such as Scrum, to the adherence to formal quality reference models such as SEI-CMMI™. But the evidence shows that very few of the organizations are keen to embrace the scale of investment and long-term commitment that such adherence requires, and has difficulties to figure out a reasonable business case for that move to be justified. Empirical evidence (Khurshid Bannerman Staples, 2009) supports the modeling of a situation where the scale of the organization is a key factor to achieve the critical mass required for a larger investment and long-term commitments, like the ones required to achieve higher levels under the most common maturity level reference models.

At the same time, empirical evidence seems to suggest that smaller organizations tend to embrace more generic quality frameworks such as the ISO 9001:2015 (ISO, 2020) as their choice to guide their quality process institutionalization efforts.

Although conceived as a guideline to generic assessments of quality systems on a variety of industries and activities, an effort is done to map how the different requirements of such framework can help to boost the performance of known software engineering strengths in a particular software development activity. At the same time, it is deemed useful to evaluate which aspects are not truly addressed by this approach, and, therefore, would benefit from supplemental activities and initiatives.

Until now, research work had to be done using sources from different development markets and results extrapolated to Argentina under the implicit assumption of validity.

This paper proposes a contribution by performing an initial analysis of a mapping between the ISO 9001:2015 requirements and its alignment with strong software engineering practices provided by agile frameworks in general and SCRUM in particular.

## **2 ISO 9001 as a quality framework for software development**

The ISO 9001 standard is very well-known in the industry at large. According to the American Society for Quality (ASQ), organizations use the standard to demonstrate the ability to consistently provide products and services that meet customer and regulatory requirements. It is the most popular standard in the ISO 9000 series and the only standard in the series to which organizations can certify. The ISO 9001 is not industry-specific and can be applied to organizations of any size. Software companies, like any other industry, can use the standard, and there is a guideline (ISO 90003, 2018) that provides guidance for software organizations in the application of ISO 9001:2008.

In Argentina, the ISO 9001 standard is widely accepted by software development companies. In the 2018 annual survey among members of the CESSI chamber, (published in April 2019), over 60% of the respondents indicated as having an ISO 9001 certification (Observatorio Permanente de la Industria de Software y Servicios Informáticos de la Argentina, 2019). The main reason for this percentage is likely the incentives yielded by the so-called *Ley de Economía del Conocimiento* (Ley 27506), which provides tax reductions, among other benefits, for those companies achieving a quality certification among other requisites.

### 3 Relationship between Agility, Scrum and ISO 9001 requisites

Stålhane and Hanssen (Stålhane & Hanssen, 2008) outlined a map between the ISO 9001 requirements and agile practices, showing that a very good fit can be identified, and also performing a preliminary overview of items that need a further and deeper analysis for a full ISO 9001 compatibility. The experience paper (Ruiz de Mendarozqueta & Oliva, 2016) comments on two implementations of agile concepts and Scrum at two companies as their strategy for obtaining their ISO 9001 certificates; their approach using a high-level map between ISO requirements to fulfill agility and SCRUM requirements.

### 4 Map between Agility, Scrum, and ISO 9001 requisites

The research question is to explore up to which extent agile principles and Scrum have strong and comprehensive coverage of ISO 9001 requisites. In order to achieve that, a high-level map between Agile principles and Scrum compared with the ISO 9001:2015 document sections will be performed. An actual ISO 9001 certification will require additional activities and proof of institutionalization mechanisms, therefore such a map is not a guarantee for getting an ISO 9001 certification nor is it an exhaustive mapping between possible relationships. It is meant to be a help to create a roadmap in that direction.

#### 4.1 Agile Manifesto as a policy

The Agile Manifesto (Beck Cockburn Fowler et al, 2001) is a document initially written in 2001 by 17 software development experts proposing a breakthrough change in the ways to develop software aiming at deconstructing the excessively rigid formalisms and focusing on customer needs instead. The main drive was to outline four values that, since then, have been inspiring the different flavors of agile methodologies. The complete documentation has been formalized by Beck (Beck, et al., 2001), some evolution has been documented by (Duncan, 2019).

The manifesto is complemented by 12 principles that highlight some fundamental ground rules such as customer integration in the development process, ownership by the entire team of everything that is produced, and a sustainable pace of work.

In brief, the dominant principles are:

1. *Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.*
2. *Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.*
3. *Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.*
4. *Business people and developers must work together daily throughout the project.*
5. *Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*
6. *The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*
7. *Working software is the primary measure of progress.*
8. *Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.*

9. *Continuous attention to technical excellence and good design enhances agility.*
10. *Simplicity --the art of maximizing the amount of work not done-- is essential.*
11. *The best architectures, requirements, and designs, emerge from self-organizing teams.*
12. *At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.*

The Agile Manifesto can be considered as the policy for establishing Agile. In particular, this paper focuses on Scrum as the agile methodology of choice given the widespread acceptance by Argentina's organizations as confirmed by previous research activities performed.

The Scrum approach institutionalizes activities using 5 key ceremonies which are:

- Backlog grooming (product backlog refinement)
- Sprint planning.
- Daily scrum.
- Sprint review.
- Sprint retrospective.

#### 4.2 ISO Section to Agile principle map

Follows an analysis between sections of the ISO 9001 framework where a stronger relationship with Agile and SCRUM can be found as compared with the Agile Manifesto. Only sections that refer to requirements for operationalization are considered. In that regard, sections 0 to 4 are deemed as infrastructure needs of the standard and thus constitute a non-functional statute. The high-level mapping can be seen at Table 1.

Development Area	ISO9001:2015 Section	Agile principle
Leadership	5.1 Leadership and commitment	Management participation Leadership key to agile success
	5.1.2 Customer Focus	Agile principle 1 to 4
	5.2 Policy	A clear statement of policy (12 principles)
	5.3 Organizational roles, responsibilities, and authorities	Overall agile management approach
Planning	6.1 Actions to manage risks and opportunities	5 key ceremonies
	6.2 Quality goals	
	6.3 Change management planning	
Support	7.1 Resources	Agile principles 5 and 6
Operational planning and control	8.1 Operational planning and control	Scrum master cycle & ceremonies
Requirements for products and services	8.2.1 Customer communication	Agile principle 1 et al.
	8.2.2 Determining the requirements related to products and services	Product owner participation Scrum ceremonies
	8.2.3 Review of requirements	
	8.2.4 Changes for products and services	Backlog prioritization Agile principle 2

Design and development	8.3 Design and development of products and services	Agile principle 9 to 11
Performance	9.1 Measurement, analysis and evaluation	5 Scrum ceremonies, some additional activities might be required.
	9.2 Internal audit.	
	9.3 Upper management review.	
Continuous Improvement	10.2 Non-conformance correction.	5 Scrum ceremonies in particular daily scrum and retrospective ceremony
	10.3 Continuous improvement	
Products and Service Management	8.5 Production and service provision	Agile principle 5 Scrum lifecycle Sprint Ceremonies
	8.6 Release of products and services	
	8.7 Control of nonconforming	

Table 1 High-level map between ISO9001:2015 and Scrum

ISO 9001:2015 Sections 0 to 4 are part of the framework infrastructure as non-functional requirements and thus there is no need for a mapping to be performed.

#### 4.3 Leadership

In Lean philosophy (Poppendieck & Poppendieck, 2003), managers must apply the Lean principles and also must be the teachers who educate the staff in the principles of the Lean philosophy. In turn, they must go to the place where things are done to understand them, in the same place they happen. The Agile manifesto (Beck, et al., 2001) are the principles that the management must establish and promote in the company. Leadership is a key aspect in agile success (Cohn, 2010) (Cohn, 2012) (Cockburn, 2007).

Regarding policy, the agile approach prioritizes individual actions and their interactions over process and tools, leverage the software as documentation, cooperation, and close teamwork with the customer (represented by a product owner) above negotiation and, perhaps the most significant component, incorporate change into the methodology rather than opposing it following a pre-defined plan.

Finally, to implement customer focus the Agile Principles number 1 to number 4 (Beck, et al., 2001), stated a strong customer focus to avoid the common pitfalls in software projects, reduce risk and deliver value and high-quality software.

The organizational roles, responsibilities, and authorities are clearly defined in Scrum as roles and responsibilities for its execution and performance which satisfies this requirement (SCRUMstudy, 2013) (Cockburn, 2007) (Cohn, Succeeding with Agile, 2010).

#### 4.4 Planning

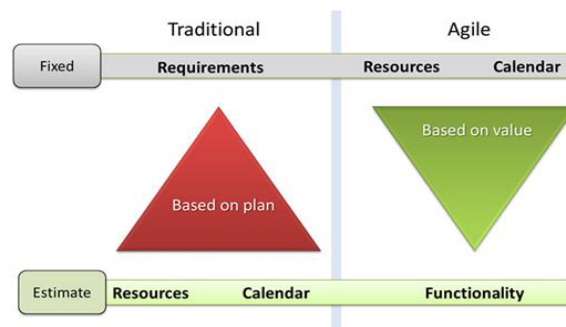
Planning is performed at a high level in the planning ceremony and daily meetings. This addresses key principles 1 through 7 with enough evidence being collected to assess compliance with the planning activities, risk management, and change planning.

#### 4.5 Support

Regarding resources, agile principles 5 and 6 have the same objective as the ISO 9001 7.1 section for providing infrastructure and an adequate environment for the people performing the activities.

#### 4.6 Operational planning and control

This requirement from ISO 9001 is covered in the Agile paradigm (Cockburn, 2007), as shown in Figure 1 Agile conceptual modeling (Morse, 2012) (Morse, 2012) because a fixed time window is established, a small team of developers is organized and functionality is continuously evaluated, with the permanent help of the "owner" of the requirements providing the necessary sponsorship. The team planned which requirements are going to be delivered at the end of the time window.



*Figure 1 Agile conceptual modeling (Morse, 2012)*

Backlog Planning and Sprint Planning activities inside Scrum (SCRUMstudy, 2013) determine the planning activities.

#### 4.7 Requirements for products and services

Customer communication requirements are addressed by almost all agile principles. This is due to the fact that team communication, empowerment, and feedback are key features of agility, and, at the same time, they form a strong basis for customer communication. Those principles are instantiated in Scrum roles and activities, such as Product Owner, Sprint Planning, Backlog Planning, and Sprint Review, as described by (SCRUMstudy, 2013) (Schwaber & Sutherland, 2017).

The requirements definition and review are performed during the different ceremonies in Scrum (Schwaber & Sutherland, 2017) activities for requirements definitions, management and implementation are clearly stated with artifacts: Product Backlog, Sprint Backlog, roles: Product Owner and ceremonies: Sprint Planning, Backlog Planning and Sprint Review.

The critical activity of managing changes in requirements is addressed by agile principle number 2 (Beck, et al., 2001). Scrum, as per (Schwaber & Sutherland, The Scrum Guide, 2017), addresses the requirements definition, management, and implementation of changes, by means of artifacts like Product Backlog and Sprint Backlog, roles like Product Owner, and ceremonies like Sprint Planning, Backlog Planning and Sprint Reviews.

#### 4.8 Design and development of products and services

The critical technical activities involved with the design and development are addressed at a high level by agile principles number 9, 10, and 11.

The design and development starts with software architecture and design, both of which determine software behavior and quality attributes. If such behavior and attributes satisfy the customer requirements, the agile process is delivering valuable software as stated in Principle number one.

On top of that, agility aims to eliminate waste (Poppendieck & Poppendieck, 2003). Considering the work partially done, the extra features and the defects, as sources of waste, all of which a good design tends to minimize, it is clear that agility has a strong focus towards good technical design and development as a source of value.

A particular note could be made regarding documentation. As stated in the Agile Manifesto, (Beck Cockburn Fowler et al, 2001), working software has more value than comprehensive documentation, therefore the Agile approach towards documentation could be considered austere, attempting to embed in the software itself, the hints of use and administration that traditionally would belong to user documents.

On the other hand, sound design and good code tend to be self-documented, being possible to deploy practices that allow for the automatic extraction of navigable software documentation from the code itself.

#### 4.9 Product and service management

In Scrum (Schwaber & Sutherland, 2017) there is an activity called *Sprint Review* where the Scrum Team and stakeholders jointly reflect on what was done in the Sprint. The done criteria are checked and the finished work is analyzed for potential release. If an item shows failures or is not conforming to what is expected by the stakeholders, the Product Backlog and Sprint Backlog are modified in order to cope with those nonconformities.

The Agile Principles number 5 is strong guidance for assuring that the product or service implements requirements, fulfills the done criteria and avoids nonconformity.

#### 4.10 Performance

As part of the 5 key ceremonies (Schwaber & Sutherland, 2017), the retrospective is the one devoted to analyzing performance during past Sprint and finding opportunities for improvement in the forthcoming one,

In this instance, the team evaluates the metrics collected, the backlog status and the velocity reached, which ultimately expresses the team's capability to deliver the required software features, within the expected calendar, cost, and quality level.

However, this activity does not fulfill the requirements expressed in section 9.2 (internal audit) which usually requires an independent view that is not explicitly contained in the Scrum methodology and needs to be instantiated separately.

In this sense, previous work (Gislén, 2016) shows that it can be achieved by having internal and external auditors participate in ceremonies, and by tailoring ISO terminology within the meetings, albeit not in the audit reports (e.g. Hit/Miss agile terminology for ISO 9001 terminology Major/Minor Non-Conformity, Observation or Noteworthy efforts)

A similar situation happens with section 9.3 (upper management review), as the scrum defines the team as self-contained being the product owner the maximum authority. In practice, however, the organizations might be structured around higher management layers than the product owner. These upper management levels are normally kept informed, as a matter of fact, however, this information flow is not explicitly required by agile processes, and, therefore, needs to be formally instantiated to satisfy requirements from ISO9001.

A reasonable means to achieve that is by having the product owner act as a proxy for the upper management, we believe that this role is well suited for the task because it has a clear notion regarding the degree up to which the customer requirements are satisfied as the product evolves sprint after sprint.

#### **4.11 Continuous Improvement**

As part of the 5 key ceremonies (Schwaber & Sutherland, 2017), in particular, the retrospective-based activities opportunities to collect and analyze performance data is given to the team towards the fulfillment of the agile principle 12. This information can be used together with proper process improvement methodologies to perform adjustments in the team performance and delivery across different sprints.

### **5 Discussion**

The results shown by the previous analysis at the conceptual, bibliographic and systemic dimensions, although preliminary, seem to be pretty consistent with the practical experience of the authors in real-world projects of different sizes and complexities where, more often than not, the projects were old fashioned Software Engineering fundamentals are not enforced, the technical debt increases with the successive sprints eroding customer trust in the new features incrementally delivered, generating schedule overruns at a product level, and forcing to add extra effort, and hence cost, in the form of additional sprints whose backlog is mainly composed of defect-correction stories. More often than not, the actual investment the software project enables is highly leveraged with a much bigger investment return, and therefore, the entire investment is jeopardized if the delivery excellence is not carefully managed. In addition to that, the effort consumed by sprints devoted to defect correction stories is essentially waste, contradicting, therefore, the Agile principle that states that “Simplicity – the art of maximizing the amount of work not done, is essential”. The author’s experience shows that in order to fulfill at product level the Agile principle that “working software is the primary measure of progress”, certain practices and metrics borrowed from the plan-driven software engineering processes may be relevant to be exercised.

The evaluation under a formal quality framework is in this regard perceived as providing organizations with two distinctive outcomes; on one hand, it provides a formal token of the rigorous implementation of certain requisites collectively deemed as a quality system; on the other hand, it helps the organization to embrace mature practices for the management of their business. Organizations, especially SME, are often willing to embrace agile concepts as a way to improve their technical delivery capabilities and customer satisfaction, but the connection between agile practices are not always linked with the actual execution of software engineering practices, and, even when connected, are not perceived as good enough to aim for formal evaluations or certifications.

Several bibliography references and the previous work from the authors build a conceptual roadmap on the relevance of agile methodologies as the conduit for the adoption of software engineering practices, the importance of protecting the value yielded by the agility by using best practices, still, the survey of



Argentinean organizations suggests that the preferred quality framework form SME sized organizations is still ISO9001. Little effort has been made to map how rigorous execution of agile methodologies can at the same time address the requirement of such framework and thus help not only to execute but also to assess in a structured way the operations.

The map we developed shows how the different major requirements of ISO9001 correlate to agile activities that need to be done and understood by the team. Metrics collection on topics other than velocity and crump down-related evolutions needs to be introduced as well. Two exceptions are identified as part of this map, one related to the need to introduce an independent view on the team operation (internal audit) and mechanisms for the upper management to be explicitly being kept informed. The authors believe that the definition of practices and collection of these additional activities shall be as agile as the rest of the process

## 6 Future work

Further work is needed to develop ideas toward a framework following the line of work of the I+D effort this paper is part of, including the identification of prototype projects where factual data can be extracted for further validation of the premises, as well as to collect metrics enabling the comparison actual evaluation activities, either formal or informal, against the ISO9001:2015 framework. Also, further characterization of the emergent trend to apply hybrid approaches, as an example using combinations such as Kanban with Scrum, to software development in terms of mixtures between agile and Software Engineering process models is needed. Of particular interest are projects at some larger scale, where the importance of uncovering, understanding and effectively applying best practices and a solid framework will be increasingly important for practical purposes at international markets and, as such, a topic for further relevant research work.

## 7 Bibliography

- Agile Alliance. (2020, 08 12). *Agile Practices Timeline*. Retrieved from Agile Practices: <https://www.agilealliance.org/agile101/practices-timeline/>
- ASQ. (n.d.). *ISO*. Retrieved 9 4, 2021, from WHAT IS ISO 9001:2015 – QUALITY MANAGEMENT SYSTEMS?: <https://asq.org/quality-resources/iso-9001>
- Beck Cockburn Fowler et al. (2001). *Agile alliance*. Retrieved from <https://www.agilealliance.org/agile101/the-agile-manifesto/>
- Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Principles behind the Agile Manifesto*. Retrieved from <http://agilemanifesto.org/principles.html>
- Cockburn, A. (2007). *Agile Software Development*. , Addison-Wesley.
- Cohn, M. (2010). *Succeeding with Agile*. Addison Wesley.
- Cohn, M. (2012). *Essential Scrum*. Adisson Wesley.
- Duncan, S. (2019). *Understanding Agile Values & Principles*. C4Media, InfoQ.com.
- Gey Fried Langer. (2015). Certification matters competition of market rational-bureocratic and professional logics in SW development organizations. Linkoping: Sweden.
- Gislén, M. (2016). *Achieving Agile Quality - An Action Research Study*. Karlskrona.

- Good, J. M. (2003). *A Pragmatic Approach to the Implementation of Agile Software Development Methodologies in Plan-Driven Organisations (MSc Thesis)*. s.l. Lincoln University.
- ISO. (2020, 06 08). *ISO 9000:2015*. Retrieved from <https://www.iso.org/obp/ui/es/#iso:std:iso:9000:ed-4:v1:es>
- ISO 90003. (2018). *ISO 90003 Directrices para la aplicación de la IRAM-ISO 9001:2008 al software*. CABA: IRAM.
- Johnson, B. (1995). Return on Investment (ROI) from Software Process Improvement as Measured by US Industry. *Crosstalk*.
- Khurshid Bannerman Staples. (2009). Overcoming the First Hurdle Why Organizations Do Not Adopt CMMI. *ICSP 2009 LNCS 5543*, 38–49.
- Morse, L. (2012). *3 Paradigm Shifts of Agile*. Retrieved 05 04, 2019, from Solutions IQ: <https://www.solutionsiq.com/resource/blog-post/3-paradigm-shifts-of-agile/>
- Observatorio Permanente de la Industria de Software y Servicios Informáticos de la Argentina. (2019, Abril). *Camara Argentina de la Industria del Software - OPPSI- Reportes*. Retrieved from Reporte anual del sector de software y servicios informáticos de la República Argentina - Año 2018: <https://www.cessi.org.ar/descarga-institucionales-2330/documento2-290a5d80ccf2da8c5d3935209aa10568>
- OPSSI, O. P. (2016). *Reporte anual sobre el Sector de software y servicios informaticos de la republica argentina*. buenos aires: CESSI.
- Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit*. Addison Wesley.
- Ruiz de Mendarozqueta, A., & Oliva, P. (2016). Certificación ISO 9001:2008 en organizaciones ágiles:. *CoNaIISI*. Salta: UCASAL.
- Ruiz de Mendarozqueta, Bustos, & Colla. (2020). Agile and software engineering - an invisible bond. *Proceedings of the 50 JAIIO*. Buenos Aires.
- Ruiz de Mendarozqueta, Bustos, & Colla. (2020). Agile in practice a systemic approach. *Proceedings of the 50 JAIIO*. Buenos Aires.
- Ruiz de Mendarozqueta; Bustos; Colla . (2019). *Agile in practice a systemic approach*. Paper accepted for 48 JAIIO-ASSE 2019, to be published in 49 JAIIO-ASSE 2020.
- SAFe. (n.d.). Retrieved 06 08, 2020, from <https://www.scaledagileframework.com/agile-architecture/>
- Schwaber , K., & Sutherland, J. (2017). *Scrum.org*. Retrieved 06 31, 2020, from The home of Scrum: <https://www.scrum.org/resources/scrum-guide>
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. Retrieved from Scrum.org
- SCRUMstudy. (2013). *A Guide to the SCRUM BODY OF KNOWLEDGE*. Retrieved 05 31, 2020, from <https://www.scrumstudy.com/>: <https://www.scrumstudy.com/>
- Stålthane, T., & Hanssen, G. (2008). The application of ISO 9001 to agile software development. *The Norwegian University of Science and Technology*. Springer Professional.

## **Análisis del uso de frameworks en la estimación de la duración de proyectos web**

Yanina Medina, Gladys Dapozo

Facultad de Ciencias Exactas y Naturales y Agrimensura  
Universidad Nacional del Nordeste (UNNE)  
{yanina, gndapozo}@exa.unne.edu.ar

**Resumen**—La estimación de esfuerzo, tiempo y costos es una problemática vigente en el desarrollo de proyectos de software. Este trabajo se orienta a la estimación en el desarrollo de aplicaciones web, dado que la ingeniería web tiene características particulares que requieren de métodos de estimación propios. El objetivo es comparar los resultados de la aplicación de métodos de estimación paramétricos para calcular la duración de proyectos web, y analizar el caso particular de proyectos que utilizan frameworks. Como caso de estudio se utilizó la información obtenida en el desarrollo de un proyecto web realizado por estudiantes universitarios de la asignatura Taller de Programación I, cuyos contenidos se orientan a la programación web. Los resultados obtenidos indican que los métodos más específicos para el desarrollo web y con mayor aporte de información del contexto, generan valores de estimación más precisos, y que la adecuación de los parámetros para reflejar la utilización de frameworks podría mejorar los valores de estimación obtenidos, sin embargo, para asegurar estos resultados los métodos deberían contemplar parámetros específicos para este caso particular.

**Palabras claves**—Estimación software, Desarrollo web, Métodos de estimación paramétricos, Frameworks de desarrollo

### **1. Introducción**

En las provincias de Chaco y Corrientes, el sector de Software y Servicios Informáticos (SSI) muestra un desarrollo sostenido, apoyado por los respectivos gobiernos provinciales y/o municipales y las universidades nacionales con carreras de Informática radicadas en la región, contando con polos tecnológicos (Polo IT Chaco y Polo IT Corrientes) que nuclean a las empresas de la zona que buscan, mediante la asociatividad, lograr el crecimiento individual y conjunto; la transferencia de conocimiento, investigación y desarrollo; la búsqueda de alianzas estratégicas con organismos y universidades; y la mejora continua. En este marco, el rol de las universidades es, entre otros, promover la investigación aplicada para incrementar la producción de bienes y servicios que agreguen valor y generen fuentes de trabajo en las áreas estratégicas para el desarrollo, de modo de favorecer la extensión, integración y profundización del proceso de industrialización, el fortalecimiento del mercado interno y la ampliación del comercio exterior [1].

El grupo de Calidad de Software de la Universidad Nacional del Nordeste desarrolla varias líneas vinculadas con la calidad del software, con el propósito de contribuir al desarrollo del sector de SSI de la región, mediante el aporte de métodos y herramientas que incrementen la

calidad de los productos y procesos del desarrollo de software. Una de estas líneas, puntualmente referida a la estimación de proyectos software.

Para las empresas de software una estimación deficiente del esfuerzo y duración que conlleva un proyecto puede ocasionar incumplimiento de plazos, entrega de productos incompletos y pérdida de competitividad. En un estudio realizado en la región NEA [2] se indica que no es habitual que las empresas de software utilicen métodos sistemáticos para estimar duración y costo de los proyectos, la mayoría utiliza las técnicas convencionales de juicio de experto, por analogía y por consenso. Las técnicas paramétricas, son conocidas, pero poco utilizadas. También se observa que la gran mayoría no utiliza datos históricos al momento de realizar la estimación de un proyecto, aspecto que se torna importante para obtener resultados más precisos.

En este trabajo se propone la evaluación de métodos de estimación paramétricos para calcular la duración de proyectos web tomando como caso de estudio el desarrollo de un proyecto realizado por los estudiantes de la asignatura Taller de Programación I de la carrera Licenciatura en Sistemas de Información de la Universidad Nacional del Nordeste (UNNE). Esta información se utiliza para calcular la duración real que se utilizará como referencia para comparar los resultados obtenidos con la aplicación de los métodos seleccionados para este estudio, incorporando también el análisis del uso de frameworks en el desarrollo, y adicionalmente, como información del contexto, el grado de experiencia de los estudiantes.

Para lograr los objetivos propuestos se utilizó una metodología similar a la que se describe en [3], pero considerando que han surgido nuevas técnicas y herramientas para el desarrollo de aplicaciones web, se analiza en particular la utilización de frameworks enfocados en soluciones web. En línea con lo expuesto en [4] que sostienen que los estudios de replicación aumentan la confianza en los resultados anteriores cuando los hallazgos son similares cada vez, y ayudan a madurar el conocimiento abordando aspectos de validez tanto internos como externos.

En la próxima sección se presenta el estado del arte. En la sección 3 se detalla la metodología aplicada. En la sección 4 se presentan los resultados obtenidos. En la sección 5 se presentan las conclusiones. Finalmente se agregó una sección 7 que consiste en un anexo donde se detallan los valores obtenidos en cada uno de los métodos aplicados.

## **2. Estado del arte**

### *A. Estimación de software*

Según McConnell [5] “una estimación es una predicción de cuánto tiempo durará o costará un proyecto”, constituye la base para la planificación de los proyectos. El desarrollo del software requiere de la estimación para controlar y administrar los recursos que se necesitan utilizar antes y durante el proyecto para lograr los resultados esperados.

En [6] se reconoce que, en la gestión y planificación de software, producir una estimación precisa del esfuerzo necesario para completar o mantener un proyecto es de gran importancia y preocupación. Sin embargo, es difícil realizar una estimación realista en la etapa inicial del desarrollo de software, ya que la información disponible en esa etapa suele ser incompleta e incierta. Aunque la construcción de modelos formales de estimación del esfuerzo de software comenzó en los primeros tiempos de la industrialización de la producción de software, el juicio de los expertos sigue siendo la estrategia dominante para la predicción del esfuerzo en la práctica, donde la precisión de la estimación es sensible a la experiencia del profesional y, por lo tanto, es propensa al sesgo.

Carbonera [7] considera que el campo de la estimación del esfuerzo de desarrollo de software tiene un impacto crucial en la presupuestación y la planificación de proyectos en la industria. Para aportar información sobre la literatura actual presenta un estudio de mapeo sistemático sobre la estimación del esfuerzo. Los resultados recopilados indican que los enfoques de estimación de esfuerzo múltiple se utilizan con más frecuencia que uno solo, más del 90% de los estudios tuvieron a los estudiantes como los participantes más comunes en su evaluación empírica. Este artículo informa sobre desafíos que vale la pena investigar, con respecto al uso de la carga cognitiva y la interacción en equipo.

#### *B. Métodos de estimación*

Existen diversos métodos para estimar el esfuerzo de desarrollo, que son clasificados en dos categorías: la primera comprende a los métodos paramétricos, son aquellos en los cuales el proceso de cuantificación del resultado está basado en un proceso mecánico, por ejemplo, la aplicación de una fórmula derivada de los datos históricos. La segunda categoría, comprende a los métodos heurísticos, son aquellos en los cuales la cuantificación del resultado se produce a partir del juicio y/o la experiencia de un experto [8]. En un trabajo previo de los autores se elaboró un cuadro para resumir los principales métodos reportados en la literatura [9].

Los Puntos de Función (PF) constituyen una métrica para establecer el tamaño y complejidad de los sistemas informáticos basada en la cantidad de funcionalidad requerida y entregada a los usuarios. Miden el tamaño lógico o funcional de los proyectos o aplicaciones de software basado en los requerimientos funcionales del usuario. Los proyectos de ingeniería web adoptan normalmente el modelo de proceso ágil. Por este motivo, es frecuente utilizar una medición de puntos de función modificada en conjunto con los pasos de la estimación en proyectos ágiles. Algunos métodos basados en puntos de función fueron utilizados en [3] y que en esta publicación volvieron a validarse. Estos son FP-Lite, Puntos de Casos de Uso (UCP) y WebMo.

La propuesta de FP-Lite deriva del método de análisis de puntos de función definido por el Grupo Internacional de Usuarios de Puntos de Función (IFPUG), simplificando los cálculos para obtener los puntos función -que requerían de mucha documentación que no siempre está disponible-, y dejando como opcional la aplicación del factor de ajuste, dado que el aporte a la precisión que otorga no compensa el esfuerzo que se necesita para calcularlo. Este método puede ser aplicado tanto a proyectos en desarrollo o proyectos de mantenimiento [10].

La metodología UCP se basa en la utilización de casos de uso como dato de entrada para calcular el esfuerzo -en horas-hombre (hh)- necesario para el desarrollo de un proyecto de software [11]. En un intento por dar respuesta a las diferencias entre las estimaciones de desarrollos clásicos y de proyectos web, aparece WebMO, un modelo de estimación orientado a aplicaciones web basado en COCOMO II, propuesto por Reifer [12].

En la literatura se observan nuevas propuestas como ser los Objetos Web Revisados (RWO), siendo esta una actualización del método WO, cuyo objetivo es dar cuenta de los nuevos estilos y tecnologías de desarrollo web. También introduce una clasificación inicial de aplicaciones web de acuerdo con su tamaño, alcance y tecnología, para refinar aún más su estimación de esfuerzo [13]. Otro nuevo enfoque de medición de tamaño es el Factor de Complejidad Web (WCF) para obtener medidas de tamaño para aplicaciones web. WCF tiene cinco factores de medición y es el primer y principal componente para desarrollar un modelo de estimación de esfuerzo para la estimación de esfuerzo web [14]. Según [15] Web objects (WO) es considerada como la primera métrica para estimar tamaño de una aplicación web

desarrollada por Donald, J. Reifer in 2000. WO es una versión extendida de los FP convencionales, obtenidos después de agregar cuatro componentes específicos de web.

### C. Frameworks

Los frameworks de desarrollo tienden a agilizar el desarrollo dado que, utilizan un conjunto de clases que engloban el diseño para resolver un conjunto de problemas comunes existentes en el desarrollo de sistemas. Además, definen elementos imprescindibles para aplicaciones como la automatización de vistas y acceso a datos [16].

La reutilización de componentes, código y automatización de aspectos del desarrollo de sistemas van dirigidos a eliminar el tiempo y el esfuerzo necesarios para construir componentes de software. Este aspecto permite que el desarrollador pueda dedicar más tiempo a tareas y requisitos funcionales, además de centrarse en mantener un código de alta calidad y sostenibilidad [17, 18].

Existen frameworks que dan soporte a diferentes lenguajes de programación para construir aplicaciones web. Según [19], existen los llamados frameworks para PHP, como por ejemplo Kumbia, Wasp, CodeIgniter, Seagull, BlueShoes, Qcodo, Akelos, PhpOpenbiz, Zoop, Ash.MVC, Diy, Wact, Zend Framework, CakePHP, Symfony, Yii, Laravel. En su trabajo, el autor los analizó y comparó con el fin de brindar a los usuarios una idea de cuál elegir para satisfacer sus necesidades, a la hora de realizar tareas con frameworks en lenguaje PHP.

El trabajo que se describe en [20], evalúa varios frameworks de código abierto con el fin de encontrar el más adecuado para el desarrollo de un sistema de apoyo a la toma de decisiones. Los frameworks seleccionados fueron: Rails (Ruby), Django (Python), Grails (Java) y Play (Java).

Según [21], los avances en la tecnología web, en los últimos años han impulsado a los desarrolladores de software a programar aplicaciones móviles responsivas amigables. Las aplicaciones web pueden fácilmente convertirse en aplicaciones móviles y por este motivo, los productos de software se desarrollen continuamente a un ritmo mucho más rápido con nuevas funciones añadidas a diario. Por ejemplo, HTML5 que evoluciona de HTML e incluye nuevos atributos y comportamientos. Aparte de HTML5, los componentes básicos de la mayoría de los navegadores modernos incluyen JavaScript (JS) y hojas de estilo en cascada (CSS3).

Respecto al framework Bootstrap es el marco HTML, CSS y JS más popular para desarrollar proyectos responsive y móviles en la web. Es una biblioteca de código abierto de componentes de UI desarrollada por Twitter. Los componentes están contruidos utilizando los principios de diseño web responsivo, lo que hace que esta biblioteca sea extremadamente valiosa para aplicaciones web que necesitan ajustarse automáticamente a su diseño en función de la resolución de la pantalla.

Los autores de [22], luego de que evaluaran la relación que existe entre el uso de un *framework* y la generación de código fuente, concluyen que la mayoría de los métodos de estimación no contemplan la influencia de las herramientas de desarrollo en la generación de código de forma automática; pudiendo dar valores alejados a la realidad.

## 3. Metodología

En este trabajo se presentan los resultados de la aplicación de 3 (tres) métodos de estimación paramétricos para calcular la duración de proyectos web tomando como caso de estudio el desarrollo realizado por los estudiantes de la asignatura Taller de Programación I de la carrera Licenciatura en Sistemas de Información, en cuyo desarrollo se utilizaron frameworks. El objetivo es comparar los resultados de la aplicación de métodos de estimación

paramétricos para calcular la duración de proyectos web, y analizar el caso particular de proyectos que utilizan frameworks.

La metodología se compone de los siguientes pasos:

- Descripción del caso de estudio.
- Cálculo de la duración real del proyecto web desarrollado por los estudiantes
- Estimación de la duración del proyecto con métodos paramétricos
- Comparación de los resultados obtenidos con cada método.
- Ajuste de los parámetros de Webmo para analizar los valores estimados consideran el uso de frameworks y la experiencia de los desarrolladores

*D. Descripción del caso de estudio.*

Se denomina caso de estudio al proyecto web que los estudiantes de la asignatura Taller de Programación I de la Licenciatura en Sistemas de Información de la UNNE. Este taller tiene como objetivo que los estudiantes adquieran las técnicas y habilidades necesarias para el desarrollo web, teniendo en cuenta componentes de calidad. Para la aplicación de las herramientas, técnicas y metodologías que conforman los contenidos de la asignatura realizan un proyecto de desarrollo, siguiendo las consignas dadas por los docentes. Con la información aportada por los estudiantes se calcula un valor de “duración real” aproximada con la cual se contrastarán los resultados de duración estimados con los distintos métodos.

El proyecto se desarrolla a largo del cursado (4 meses), se trata de una aplicación web sencilla pero completa, que incluye todos los componentes necesarios: modelado de la aplicación, diseño gráfico y de contenidos, gestor de base de datos, tecnologías de programación en cliente y en servidor. Si bien los estudiantes realizan el desarrollo en forma individual, las características y complejidad del proyecto son las mismas para todos. Esta información se utiliza para asignar los valores de los parámetros de los distintos métodos.

Las consignas dadas a los estudiantes en el cursado 2021, son las siguientes:

- Desarrollar una aplicación orientada al comercio electrónico, respetando los criterios vinculados con la presentación y las funcionalidades requeridas, que fueron previamente definidos por el profesor.
- Realizar la especificación de requerimientos utilizando el estándar de IEEE 830, que incluye el modelado con diagramas de casos de uso.
- Utilizar los frameworks Bootstrap y CodeIgniter, los lenguajes Html5, CSS3 y PHP5; y PhpMyAdmin como gestor de datos.
- Realizar la presentación de los resultados del proyecto en 2 fases: la primera incluye la programación en el framework Bootstrap, y la segunda, la programación en el framework CodeIgniter, diseño y conexión de la base de datos.
- Por cada fase, el alumno debe llevar un registro del desarrollo completando un formulario provisto al efecto [23], indicando la tarea realizada y la duración correspondiente.

Además, como información complementaria, se les solicitó indicar el grado de experiencia en programación en un formulario Google docs [23], indicando su nivel de experticia en programación, de acuerdo con las siguientes categorías:

- i. En los últimos 3 años se ha desempeñado como programador en áreas de sistemas o empresas de software.
- ii. No tiene experiencia laboral en programación, pero programa con solvencia, habiendo realizado capacitaciones extracurriculares.
- iii. La experiencia de programación que posee es la de las asignaturas de programación de la carrera, en las que no tuvo problemas.

- iv. Siempre le ha costado la programación y esta es la primera vez que desarrolla un proyecto completo

Al final del plazo indicado, los estudiantes presentaron 15 proyectos de desarrollo individual. Por cada proyecto, se consideró la duración total del mismo, tomando la duración reportada por el propio alumno en el formulario mencionado.

*E. Cálculo de la duración real del proyecto web desarrollado por los estudiantes*

Con la duración consignada por cada estudiante se calculó una duración real como el promedio de las duraciones reportadas por los estudiantes.

*F. Estimación de la duración del proyecto con métodos paramétricos*

Se realizó la estimación de la duración aplicando los métodos FP-Lite, Casos de Uso y Webmo, considerando las características del proyecto realizado, descriptas en el apartado A, las herramientas utilizadas y las particularidades del contexto del desarrollo. Se obtuvieron los valores de duración estimados resultantes de cada método.

*G. Comparación de los resultados obtenidos con cada método.*

Los valores estimados se compararon con la duración real calculada.

*H. Ajuste de los parámetros de Webmo para analizar los valores estimados consideran el uso de frameworks y la experiencia de los desarrolladores*

Considerando que el método Webmo es un modelo de estimación orientado a aplicaciones web basado en COCOMO II, propuesto por Reifer [12], considera atributos específicos de este tipo de desarrollo, se propone ajustar los valores de algunos parámetros en los que incidiría el uso de frameworks. Adicionalmente, dado que considera datos propios del contexto, se evalúa también el impacto de la experiencia de los desarrolladores.

## 4. Resultados

A continuación, se presentan los resultados de cada una de las etapas:

*B-Cálculo de la duración real del proyecto*

Al finalizar el plazo previsto, se presentaron 15 proyectos. A partir de los datos obtenidos del formulario destinado al registro de las actividades de cada estudiante, se obtuvo un valor de duración real del desarrollo del proyecto que surge del promedio de las duraciones informadas.

Para la primera fase se estableció una duración promedio de 2 hs por día, durante 14 días. Y para la segunda fase una duración promedio de 4 hs por día, durante 30 días. Considerando la duración medida en meses de 20 días de 5 hs, se obtuvo una duración real de **1,53** meses.

Duración:  $(A+B+C+D) / 5$

A= Promedio de horas en la etapa de análisis

B= Promedio de horas en la etapa de diseño

C= Promedio de horas en la etapa de codificación

D= Promedio de horas en la etapa de prueba

$(2,26 + 1,73 + 1,73 + 1,93) / 5 = \mathbf{1,53}$  meses



*C-Estimación de la duración del proyecto con métodos paramétricos, ajustando sus parámetros para reflejar el uso de frameworks.*

En el Anexo (Ítem 7), se muestran los resultados de la aplicación de cada uno de los métodos considerando las características del proyecto. Cabe mencionar que en todos los casos se consideró el mes de 20 días y 5 hs por día, por la naturaleza de un trabajo académico, en el cual los estudiantes no tienen una dedicación exclusiva al proyecto.

*D- Comparación de los resultados obtenidos con cada método.*

En la tabla 1 se presentan los valores obtenidos y se observa que las estimaciones obtenidas son coherentes con las características de los métodos. Los puntos de casos de uso son un método de estimación temprana con lo cual no incorpora datos que puedan aportar mayor precisión. El método FP-Lite utiliza principalmente las funcionalidades básicas del sistema aportando parámetros en función de las características técnicas del desarrollo y Webmo aporta información específica del contexto de desarrollo de proyectos web.

**TABLE I.** DURACIONES OBTENIDAS VS DURACIÓN REAL

Duración real	Casos de uso	FP-Lite	Webmo
1,53	3,97	2,7	2,7

*E-Ajuste de los parámetros de Webmo para analizar los valores estimados consideran el uso de frameworks y la experiencia de los desarrolladores*

Para considerar el uso de framework en los multiplicadores de esfuerzo se modificó el valor del parámetro FCIL, relacionado con las facilidades en relación con la disponibilidad de herramientas, al valor **Veryhigh** de acuerdo con las categorías establecidas en [13]. Se obtuvieron los siguientes valores expresados en la tabla 2.

**TABLE II.** SELECCIÓN DE MULTIPLICADORES DE ESFUERZO

Cost Driver	Ratings				
	Verylow	Low	Nominal	High	Veryhigh
CPLX	0,63	<b>0,85</b>	1	1,3	1,67
PDIF	0,75	<b>0,87</b>	1	1,21	1,41
PERS	1,55	1,35	<b>1</b>	0,75	0,58
PREX	1,35	<b>1,19</b>	1	0,87	0,71
FCIL	1,35	1,13	1	0,85	<b>0,68</b>
SCED	1,35	1,15	<b>1</b>	1,05	1,1
RUSE	1,35	1,15	<b>1</b>	1,25	1,48
TEAM	1,45	<b>1,31</b>	1	0,75	0,62
PEFF	1,35	1,2	<b>1</b>	0,85	0,65

En los Objetos Web se modificó el peso de los parámetros que estarían influenciados por el uso de frameworks, tales como: EQ (archivos de consultas), MMF (archivos multimedia), WBB (bloques de construcción web, como por ejemplo componentes como Shopping cart, botones, logos) y QL (líneas de consultas, script), Los valores obtenidos se muestran en la tabla 3.

**TABLE III.** RECuento DE OBJETOS WEB

Web Objects Predictors	Cantidad x peso	Total
ILF	4x7	28
EI	9x3	27
EQ	2x3	6
MMF	4x3	12
WBB	3x3	9
QL	4x2	8
Total		90

Realizado el cálculo del método Webmo con los parámetros ajustados para reflejar el uso de frameworks, se obtuvieron los resultados que se muestran en la tabla 4.

**TABLE IV.** ESTIMACIÓN CON WEBMO CONSIDERANDO EL USO DE FRAMEWORKS

<b>Esfuerzo:</b> 203,2 personas-mes
<b>Duración total:</b> 28,5 meses (20 días, 8hs)
<b>Duración del desarrollo:</b> 11,4 meses (40% correspondiente al desarrollo)
<b>Duración del desarrollo ajustado:</b> <b>2,03</b> meses (20 días, 5 hs)

Estimación actual de la duración: **2,03** meses

Dado que en este método se contempla la información del contexto para aportar mayor precisión a la estimación, se evaluó el cálculo considerando la experiencia, tomando como base las categorías en que los estudiantes se ubicaron según su grado de destreza en programación, que se asociaron a los valores del Cost Driver PREX, de la siguiente manera:

Alto: En los últimos 3 años se ha desempeñado como programador en áreas de sistemas o empresas de software.

Nominal: No tiene experiencia laboral en programación, pero programa con solvencia, habiendo realizado capacitaciones extracurriculares.

Bajo: La experiencia de programación que posee es la de las asignaturas de programación de la carrera, en las que no tuvo problemas.

Muy bajo: Siempre le ha costado la programación y esta es la primera vez que desarrolla un proyecto completo

En la tabla 5 se presenta el cambio de valor del multiplicador de esfuerzo PREX, considerando el valor HIGH de 0,87.

**TABLE V.** SELECCIÓN DE MULTIPLICADORES DE ESFUERZO

Cost Driver	Ratings				
	Verylow	Low	Nominal	High	Veryhigh
CPLX	0,63	0,85	1	1,3	1,67
PDIF	0,75	0,87	1	1,21	1,41
PERS	1,55	1,35	1	0,75	0,58
PREX	1,35	1,19	1	<b>0,87</b>	0,71

Cost Driver	Ratings				
	Verylow	Low	Nominal	High	Veryhigh
FCIL	1,35	1,13	1	0,85	0,68
SCED	1,35	1,15	1	1,05	1,1
RUSE	1,35	1,15	1	1,25	1,48
TEAM	1,45	1,31	1	0,75	0,62
PEFF	1,35	1,2	1	0,85	0,65

Para este nuevo cálculo de estimación, se consideró el recuento de web objects indicados en la tabla 2. Los resultados obtenidos se muestran en la tabla 6.

**TABLE VI.** ESTIMACIÓN CON WEBMO CONSIDERANDO ALTA EXPERIENCIA DE LOS DESARROLLADORES

<b>Esfuerzo:</b> 148,5 personas-mes
<b>Duración total:</b> 24,3 meses (20 días, 8hs)
<b>Duración del desarrollo:</b> 9,7 meses (40% correspondiente al desarrollo)
<b>Duración del desarrollo ajustado:</b> 1,48 meses (20 días, 5 hs)

Estimación actual de la duración: **1,48** meses

Con los ajustes realizados para reflejar las características del proyecto utilizando frameworks y un nivel alto de experiencia, se puede apreciar que el valor de la estimación mejora, aproximándose a la duración real calculada, tal como se muestra en la tabla 7.

**TABLE VII.** COMPARACIÓN DE VALORES OBTENIDOS EN WEB CON AJUSTE DE PARAMETROS

Duración real	Webmo	Webmo con Framework	Webmo con experiencia alta
1,53	2,7	2,03	1,48

## 5. Conclusiones

Se aplicaron métodos de estimación paramétricos para calcular la duración de un proyecto *post-mortem*, con el objetivo de evaluar la precisión de la estimación en cada caso. Se utilizaron los métodos basados en Casos de Uso, FP-Lite basados en puntos de función y el método Webmo, este último específico para aplicaciones web. La duración calculada por Webmo y FP-Lite son las que más se aproximan a la duración real y la más alejada es la duración proporcionada por el método de Puntos de Casos de Usos, lo que indica que cuanto más información del proyecto se considere, más precisa será la estimación. De todos modos, la estimación provista por el método basados en Casos de Uso puede ser útil en etapas tempranas del proyecto, apoyando el cálculo del presupuesto del desarrollo.

En el caso particular de Webmo, específico para aplicaciones web, se requiere información de proyectos más avanzado en las etapas de desarrollo, e información histórica y del contexto del desarrollo que aportarán mayor precisión en la estimación.

Cabe destacar que en cada modelo se deben ajustar los parámetros a la situación particular del caso que se analiza, y si bien la selección de los valores puede resultar subjetiva, para las

empresas que desarrollan software utilizar un método paramétrico de estimación representa un buen punto de partida para configurar valores propios en función de la experiencia y del tipo de desarrollo, y generar información histórica que se pueda utilizar en futuros cálculos, mejorando la precisión de la estimación en ese contexto.

Considerando que el desarrollo web actual incorpora nuevas tecnologías, como por ejemplo el uso de frameworks que agilizan el desarrollo, se intentó ajustar los parámetros que establece el método Webmo para reflejar la utilización de esta tecnología, sin embargo, los resultados no muestran una mejora significativa en la estimación utilizando frameworks, lo que sí ocurre ajustando el valor del parámetro de la experiencia. Se concluye que se requiere que el método contemple parámetros específicos para el caso del uso de frameworks.

Como trabajo futuro se propone indagar sobre métodos más actualizados que contemplen específicamente el uso de frameworks en el desarrollo web.

## 6. Referencias

- [1] Cuenca Pletsch, L., Dapozo, G., Greiner, C., & Estayno, M. (2012). Vinculación universidad-empresa orientada a la promoción de la industria del software. Una experiencia de colaboración en la región NEA. Vol. 1 (2012): Revista del Núcleo de Estudios e Investigaciones en Educación Superior del MERCOSUR (versión anterior) ISSN 2313 - 9080
- [2] G. N. Dapozo, C. L. Greiner, E. Irrazábal, Y. Medina, M. Ferraro and B. Lencina, "Características del desarrollo de software en la ciudad de Corrientes", 2015. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/50415>
- [3] G. N. Dapozo, Y. Medina, B. Lencina, G. Pedrozo Petrazzini, "Análisis comparativo de métodos de estimación basados en puntos de función para proyectos web", XX Congreso Argentino de Ciencias de la Computación. (CACIC 2014). - 1a ed. - San Justo: Universidad Nacional de La Matanza, 2014. E-Book. ISBN 978-987-3806-05-6 1. La Matanza, provincia de Buenos Aires, Octubre 2014
- [4] V. Tawosi, F. Sarro, A. Petrozziello, M. Harman "Multi-Objective Software Effort Estimation: A Replication Study". IEEE Transactions on Software Engineering, 2021.
- [5] McConnell, S., Software Estimation: Demystifying the Black Art (Developer Best Practices). Microsoft, 2006.
- [6] F. Sarro, R. Moussa, A. Petrozziello, and M. Harman. "Learning from mistakes: Machine learning enhanced human expert effort estimates". IEEE Transactions on Software Engineering, 2020.
- [7] C.E. Carbonera, K. Farias, V. Bischoff, "Software development effort estimation: a systematic mapping study". IET Software, 2020, vol. 14, no 4, p. 328-344.
- [8] M. Nasir, F. Ahmad, "An Empirical Study to Investigate Software Estimation Trend in Organizations Targeting CMMI", Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse, 2006.
- [9] G. N. Dapozo, C. L. Greiner, Y. Medina, M. Ferraro, G. Pedrozo Petrazzini, B. Lencina, "Métodos de estimación de software. Un análisis desde un enfoque evolutivo." III Jornadas de Investigación en Ingeniería del NEA y países limítrofes. UTN - Facultad Regional Resistencia. ISBN: 978-950-42-0157-1. 9 y 10 de Junio de 2014.
- [10] Piattini, Medición y Estimación del Software. AlfaOmega Editores, 2008.
- [11] C. A. Remón, P. Thomas, "Análisis de Estimación de Esfuerzo aplicando Puntos de Caso de Uso." XVI Congreso Argentino de Ciencias de la Computación, 2010.
- [12] D. J. Reifer, "Web Development: Estimating Quick-to-Market Software." Software, IEEE Computer Society, November/December 2000, pp. 57-64. 2000
- [13] R. Folgieri, G. Barabino, G. Concas, E. Corona, R. De Lorenzi, M. Marchesi and A. Segni, "A revised web objects method to estimate web application development effort." In Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics (WETSoM '11). Association for Computing Machinery, New York, NY, USA, 59-64. 2011. DOI:<https://doi.org/10.1145/1985374.1985388>

- [14] S. M. Saif and A. Wahid, "Web complexity factors! A novel approach for predicting size measures for web application development," 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, 2017, pp. 897-902, doi: 10.1109/ICICI.2017.8365266
- [15] S. M. Saif and A. Wahid, "Web Effort Estimation Using FP and WO: A Critical Study," 2018 Second International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2018, pp. 357-361, doi: 10.1109/ICCMC.2018.8487472.
- [16] R. E. Johnson, "Documenting Frameworks using Patterns. ," p. 14. Accessed on: 2017
- [17] S. H. Kaisler, "Part4 Frameworks con-cepts," in Software Paradigms, I. A John Wiley & Sons, Publication, Ed., 2005.
- [18] D. C. Schmidt, "Introduction to Patterns and Frameworks," prese
- [19] F. Sierra, J. Acosta, J. Ariza y M. Salas, "Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web", Revista Investigación y desarrollo en TIC. Vol. 4 Núm. 2 (2013): Julio – Diciembre.
- [20] L. Teixeira, A.R. Xambre, H. Alvelos, H., N. Filipe & A Ramos, "Selecting an Open-Source framework: a practical case based on software development for sensory analysis", Procedia Computer Science. 64, 1057-1064, 2015.
- [21] F. Shahzad, "Modern and Responsive Mobile-enabled Web Applications", The 12th International Conference on Future Networks and Communications. Procedia Computer Science Volume 110, 2017, Pages 410-415.
- [22] D. Santana Ballate, F. Arteaga Céspedes, V. D. Muñoz Castillo, A. Hernández González, "Evaluación del impacto del uso de un framework en la estimación del esfuerzo de desarrollo del software", Revista Cubana de Ingeniería. Vol. X, Nº 1, enero - abril 2019, pp. 69 - 78
- [23] Formulario "Taller de Programación I- Estimación web". Disponible en: <https://bit.ly/3ExgyXK>

## 7. Anexo

### *FP-Lite*

Para la estimación, este método considera el tipo de proyecto a desarrollar. En este caso, dado el contexto específico, los proyectos se consideraron nuevos, multiplataforma, utilizando lenguajes de 4ta. generación. Para el conteo de puntos de función, se analizaron los elementos funcionales requeridos, determinando los siguientes:

- ILF (ficheros lógicos internos): usuarios, productos, ventas, consultas. Total: 4
- ELF (ficheros lógicos externos): no tiene. Total: 0
- EI (entradas externas): agregar productos, agregar usuarios, eliminar productos, eliminar usuarios, modificar usuarios, modificar productos, registrar consultas, eliminar consultas, registrar facturas. Total= 9.
- EQ (consultas externas): consulta de usuarios, consulta de servicios. Total= 2
- EO (salidas externas): no tiene.

La estimación que provee es puntual, pero se recomienda utilizar un intervalo en torno al 20% del valor obtenido. El resultado optimista es el representado por PF-1, el esperado o promedio es PF-2 y el pesimista PF-3.

Dado que el método considera 6 fases para el ciclo de vida (Planificación, especificación, diseño, construcción, pruebas e implantación), en función de las características del proyecto considerado, se tomó en cuenta sólo la fase de Construcción, por tanto, se calculó el esfuerzo de desarrollo como el 40% del Esfuerzo Total, tal como se muestra en la tabla 8.

**TABLE VIII.** ESTIMACIÓN UTILIZANDO FP-LITE

	Estimación utilizando FP-Lite		
	<i>PF-1 (-20%)</i>	<i>PF-2</i>	<i>PF-3 (+20%)</i>
Total FP sin ajustar	68,8	86	103,2
Esfuerzo total (hs)	965,57	1119,77	1263,88
Duración total	3,92	<b>4,11</b>	4,28
Esfuerzo de desarrollo	386,23	447,91	505,55
Duración (meses 20d x 5h)	2,46	<b>2,72</b>	2,95

La duración se ajustó a las condiciones establecidas (mes de 20 días y día de 5 hs), y se obtuvo el valor de **2,7 meses**.

### *Puntos de Casos de uso*

Para la aplicación de este método se determinaron: Actores: administrador y cliente. Tipo: complejo porque interactúan a través de una interfaz gráfica. Casos de uso: alta, baja, modificación de productos; alta, baja, modificación de usuarios, consulta de usuarios, consulta de productos, registrar consultas de los usuarios, registrar ventas. Tipo: simple; total=10 Factores técnicos.

Utilizando la fórmula

$$TCF = 0,6 + 0,01 \sum_{i=1}^{13} TP_i Po_i, \text{ donde } TP_i \text{ es el peso del factor } i, \text{ y } Po \text{ la ponderación}$$

el valor final de TCF fue de 0,825.

En la tabla 9 se tiene un detalle de la ponderación de cada uno de los factores.

**TABLE IX.** PONDERACIÓN DE CADA FACTOR TÉCNICO DE COMPLEJIDAD

FT	Descripción	Peso	Ponderación	Total
T1	Sistema distribuido	2	0	0
T2	Objetivos de rendimiento	1	2	2
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento complejo	1	0	0
T5	Reusabilidad	1	3	3
T6	Fácil de instalar	0,5	0	0
T7	Fácil de usar	0,5	3	1,5
T8	Portabilidad	2	3	6
T9	Fácil de cambiar	1	3	3
T10	Uso concurrente	1	0	0
T11	Características de seguridad	1	2	2
T12	Provee acceso a terceros	1	3	3
T13	Formación especial requerida	1	0	0

Se considera que los factores T1, T4, T6, T10 y T13, no son relevantes (ponderación 0) dado que la aplicación no se ha diseñado para que sea distribuida, no hará cálculos matemáticos complejos, el sistema no se instala, no tendrá acceso concurrente y los usuarios no serán expertos (estará orientada para el público en general que desee comprar un artículo o contratar un servicio). Los objetivos de rendimiento y de seguridad no son prioritarios (T2 y T11), por tanto, la ponderación es 2. Los factores valorados con 3 tienen un impacto medio en el proyecto.

Respecto a los factores de entorno, con la siguiente fórmula se obtuvo el valor final de ECF de 0,5.

$$ECF = 1,4 - 0,03 \sum_{i=1}^8 EF_i Po_i, \text{ donde } EF_i \text{ es el peso del factor } i, \text{ y } Po \text{ la ponderación}$$

En la tabla 10 se muestra la valoración que recibió cada uno de los factores:

**TABLE X.** VALORACIÓN DE LOS FACTORES DEL ENTORNO

FE	Descripción	Peso	Ponderación	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4,5

FE	Descripción	Peso	Ponderación	Total
E2	Experiencia en la aplicación	0,5	3	1,5
E3	Experiencia en la orientación a objetos	1	4	4
E4	Capacidad del analista líder	0,5	4	2
E5	Motivación	1	5	5
E6	Estabilidad en los requerimientos	2	5	10
E7	Personal de medio tiempo	-1	0	0
E8	Dificultad en el lenguaje de programación	1	3	3

Los factores E1, E2 y E8 se han considerado con un valor promedio debido a la experiencia previa que adquirieron los alumnos en el transcurso de la carrera.

E3 tiene una ponderación relevante porque los alumnos previamente cursaron la asignatura Programación Orientada a Objetos y E4 porque el analista líder es el propio profesor.

E5 tiene la más alta ponderación debido a que el proyecto propone libertad para aplicar las soluciones, representa un desafío e implica la aprobación de la materia.

Igual que E6 que justifica su ponderación porque no habrá cambios en los requerimientos planteados al inicio.

E7 es 0 porque el proyecto no cuenta con personal de medio tiempo.

Se ha tomado un factor de productividad (FP) de 17 al ser casos de usos simples.

El cálculo final se presenta en la tabla 11.

**TABLE XI.** ESTIMACIÓN UTILIZANDO CASOS DE USO

<p>Puntos Caso de Uso sin ajustar (UUCP – Unadjusted Use Case Points)</p> $UUCP = \sum_{i=1}^n A_i P_i + \sum_{i=1}^m CU_i P_i$ <p>donde n es el n° de actores, m el n° de CU y <math>P_i</math> el peso</p> <p><math>UUCP = \text{Actores} * \text{Peso} + \text{Casos de uso} * \text{Peso} = 2 * 3 + 10 * 5 = 56</math></p>
<p>Puntos Caso de Uso ajustados (UCP)</p> $TCF = 0,6 + 0,01 \sum_{i=1}^{13} TFi P_{oi}$ <p>donde <math>TF_i</math> es el peso del factor i, y <math>P_{oi}</math> la ponderación</p> $ECF = 1,4 - 0,03 \sum_{i=1}^8 EF_i P_{oi}$ <p>donde <math>EF_i</math> es el peso del factor i, y <math>P_{oi}</math> la ponderación</p> <p><math>UCP = TFC * ECF * UUCP = 0,835 * 0,5 * 56 = 23,38</math></p>
<p><b>Esfuerzo</b> = <math>UCP * FP = 23,38 * 17 = 397,46</math> hh</p>
<p><b>Duración</b> = <math>\text{Esfuerzo} / \text{horas por día} / \text{días por meses} = 397,46 / 5 / 20 = 3,97</math> meses</p>

Estimación actual de la duración: **3,97** meses.



### WebMO

Para este método de estimación se ha determinado que los proyectos son de tipo comercio electrónico, por lo que los valores de las constantes de esfuerzo proporcionadas por WebMo son:  $A=2,3$ ;  $B=2$ ;  $P1=1,03$  y  $P2=0,5$ .

Como los alumnos programaron en diversos lenguajes (PHP, HTML5, CSS3, SQL) se consideró el default para los lenguajes de  $4GL = 20$  para la constante de lenguaje.

Para establecer el valor de los multiplicadores de esfuerzo, se tomaron los valores establecidos en la tabla Cost Driver que define el método [13].

**TABLE XII.** SELECCIÓN DE MULTIPLICADORES DE ESFUERZO

Cost Driver	Ratings				
	Verylow	Low	Nominal	High	Veryhigh
CPLX	0,63	<b>0,85</b>	1	1,3	1,67
PDIF	0,75	<b>0,87</b>	1	1,21	1,41
PERS	1,55	1,35	<b>1</b>	0,75	0,58
PREX	1,35	<b>1,19</b>	1	0,87	0,71
FCIL	1,35	1,13	1	<b>0,85</b>	0,68
SCED	1,35	1,15	<b>1</b>	1,05	1,1
RUSE	1,35	1,15	<b>1</b>	1,25	1,48
TEAM	1,45	<b>1,31</b>	1	0,75	0,62
PEFF	1,35	1,2	<b>1</b>	0,85	0,65

El recuento de objetos web (WO) se calculó seleccionando los valores que corresponden a cada aspecto del proyecto. Los valores resultantes se muestran en la tabla 13.

Tamaño =  $WO * \text{constante del lenguaje} / 1000$

Tamaño =  $98 * 20 / 1000 = 1,96$  (Miles de líneas de código)

**TABLE XIII.** RECUENTO DE OBJETOS WEB

Web Objects Predictors	Cantidad x peso	Total
ILF	4x7	28
EI	9x3	27
EQ	2x3	6
MMF	4x4	16
WBB	3x3	9
QL	4x3	12
Total	98	98

Para calcular el esfuerzo y la duración, se utilizan las siguientes fórmulas:

**Esfuerzo** =  $A (\prod c_i) (\text{tamaño})^{P1}$

**Duración** =  $B (\text{esfuerzo})^{P2}$

Donde:

- A, B son constantes multiplicativas de esfuerzo y P1, P2 exponentes, que se determinan en función del tipo de proyecto y del tamaño. Ver tabla 14.

- Cdi: Manejadores de costos.
- Tamaño: líneas de código fuente, expresado en miles. Se obtiene multiplicando el total de Web Objects por una constante asociada al lenguaje de programación.

**TABLE XIV.** CONSTANTES DE ACUERDO AL TIPO DE PROYECTO

Parámetros / Exponentes	A	B	P1	P2
Basado en el comercio electrónico	2,3	2	1,05	*
Aplicación financiera	2,7	2,2	1,05	*
Aplicación de B2B (negocio a negocio)	2	1,5	1	*
Aplicación basada en utilitarios informativos	2,1	2	1	*

(\*) 0,5 para menos de 300 puntos de objeto o 0,32 para más 300.

Para calcular el producto de los manejadores de costos (cdi) se seleccionan los valores (que se detallan en [13]), asociados a distintos aspectos vinculados con el proyecto:

- CPLX: Complejidad y confiabilidad del producto
- PDIF: Dificultad de la plataforma
- PERS: Capacidades del personal
- PREX: Experiencia del personal
- FCIL: Facilidades
- SCED: Restricciones de horarios
- RUSE: Grado de reutilización planeado
- TEAM: Trabajo en equipo
- PEFF: Eficiencia del proceso

Los Web Objects se calculan a partir de los siguiente predictores:

- ILF: Archivos lógicos internos.
  - EIF: Archivos externos
  - EI: Entradas externas
  - EO: Salidas externas
  - EQ: Consultas externas
  - MmF: Archivos multimedia
  - WBB: Constructor de bloques web
  - Scripts: número de macros, contenedores, etc
  - Links: número de referencias
- asignándole a cada uno un peso (bajo, medio, alto) según corresponda.

En la tabla 15 se muestran los valores resultantes de la estimación de esfuerzo y duración.

**TABLE XV.** ESTIMACIÓN CON WEBMO

<b>Esfuerzo:</b> 277,7 personas-mes
<b>Duración total:</b> 33,3 meses (20 días, 8hs)
<b>Duración del desarrollo:</b> 13,3 meses (40% correspondiente al desarrollo)
<b>Duración del desarrollo ajustado:</b> 2,7 meses (20 días, 5 hs)

Estimación actual de la duración: **2,7** meses.