

Identificación y Especificación de Requisitos de Calidad: Un Aporte para la Mejora del Desarrollo del Software Público.

Javier Saldarini¹, Claudio Carrizo¹, Juan P. Mansilla,¹Alberto Sánchez², Carlos Salgado², and Mario Peralta³

¹ Universidad Tecnológica Nacional – Facultad Regional San Francisco
Avda. de la Universidad 501, C.P. 2400, San Francisco, Córdoba, Argentina
{saldarinijavier, cjcarrizo, juampimansilla}@gmail.com

² Departamento de Informática Facultad de Ciencias Físico-Matemáticas y Naturales
Universidad Nacional de San Luis
Ejército de los Andes 950 – C.P. 5700
San Luis - Argentina
{csalgado, alfanego, mperalta}@unsl.edu.ar

Resumen. El Software Público es aquella solución tecnológica de software desarrollada, usada, modificada y distribuida, la cual permite facilitar el cumplimiento de objetivos gubernamentales y sociales, por el Estado, y para el Estado.

La Oficina Nacional de Tecnologías de Información (ONTI), entre otras, tiene la misión de promover y facilitar el desarrollo del Software Público, para ello desde la ONTI se promueven distintos instrumentos que asisten a los organismos para el desarrollo y mantenimiento de software, uno de ellos, es el Código de Buenas Prácticas en el desarrollo de software público, el cual consiste en una guía técnica para el desarrollo sustentable de software en la Administración Pública.

Visto el marco descrito anteriormente, es que el presente trabajo aborda la necesidad de indagar en qué aspectos los instrumentos puestos a disposición para el desarrollo del Software Público, contemplan la calidad del software.

Específicamente este trabajo tiene como principal objetivo desarrollar una propuesta que, desde la perspectiva de la calidad de productos de software, y tomando como marco de referencia lo establecido en la familia de normas ISO/IEC 25000, sirvan como soporte para la identificación y especificación de requisitos de calidad para el desarrollo del Software Público.

1 Introducción

El software forma parte de nuestras vidas, está en todos los equipos que manejamos, medios de transporte, sistemas de telecomunicaciones, aparatos médicos, sistemas de la administración pública, en el arte y en cualquier industria relacionada con el ocio y el entretenimiento [1]. Como se menciona en el párrafo anterior, las aplicaciones de software están presentes en muchas de las actividades que desarrollamos, de manera puntual y como ciudadanos, interactuamos con plataformas que nos brindan acceso a distintos tipos de servicios relacionados a la Administración Pública, ya sea para obtener información, o bien, para llevar a cabo algún tipo de trámite o pago de impuestos.

En este sentido el Estado Argentino, a través del Plan de Modernización del Estado [2], promueve distintos tipos de iniciativa para lograr la modernización de la gestión estatal con el objetivo de mejorar los servicios al ciudadano. En este contexto, el software es una de las herramientas clave para lograr los objetivos establecidos en el mencionado Plan.

Según se describe en [3], el contexto actual en cuanto al desarrollo de software en la Administración Pública, de cualquier nivel (municipal, provincial o nacional), es heterogéneo y diverso. Existen casos de dependencias que homogeneizan sus sistemas, compartiendo datos, información, e incluso sistemas. En este caso, se suelen contar con divisiones informáticas que realizan desarrollos propios. También se menciona que existen dependencias cuyo nivel de informatización es precario, donde no se desarrollan políticas de creación y/o adquisición de software; por último, existe un segundo caso que suele darse en dependencias que tercerizan el servicio informático.

El objeto de este trabajo consiste en focalizar el estudio en aquellas dependencias estatales que cuenten con divisiones informáticas, y que realicen desarrollos de software propio. Dado este contexto, se puede observar que, para todas aquellas reparticiones pertenecientes a la Administración Pública, que estén llevando a cabo el proceso de desarrollo de software, y quieran materializar esas iniciativas, existe un canal operativo y de soporte denominado Oficina Nacional de Tecnologías de Información (ONTI) [4].

Específicamente, dentro de la estructura de la mencionada Oficina existe el Equipo de Software Público, cuya misión es promover y facilitar su desarrollo. Este último término hace referencia a aquella solución tecnológica de software desarrollada, usada, modificada y distribuida, la cual permite facilitar el cumplimiento de objetivos gubernamentales y sociales, por el Estado, y para el Estado.

Dentro de este marco operativo existen distintos instrumentos que asisten a los Organismos para el desarrollo y mantenimiento de software, uno de ellos, es el Código de Buenas Prácticas en el Desarrollo de Software Público (CBPSP) [5], el cual proporciona una guía técnica para el desarrollo sustentable de software en la Administración Pública; esta guía fue publicada en abril del año 2019.

En [6] se menciona que tanto las empresas, como las administraciones públicas, son grandes consumidores de software, tanto del hecho a medida como de productos comerciales. Por lo tanto, la calidad del software debe ser un aspecto relevante para todas ellas.

La calidad del software es un concepto relativamente reciente; son muchos los esfuerzos desarrollados en este campo en las últimas décadas, en donde se han conseguido grandes avances, logrando una mejora en la convergencia con la calidad tradicional.

Si hablamos de calidad de software, se puede observar que en la bibliografía existen diversas definiciones, por ejemplo, en [7] se la define como “el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados, y de las características implícitas que se espera de todo software desarrollado profesionalmente”.

También se observa que, en ISO/IEC 25000 [8], se la define como el “grado en que el producto software satisface las necesidades expresadas o implícitas, cuando es usado bajo condiciones determinadas”.

La calidad del software puede ser abordada desde distintos enfoques, éstos pueden ser: calidad a nivel proceso, calidad a nivel de producto y calidad en uso. Para cada uno de estos enfoques, existen distintos tipos de modelos de calidad de software, los cuales pueden aplicarse de acuerdo a lo descrito en [9].

Visto el marco descrito anteriormente, y con el fin de realizar aportes en pos de la mejora de la calidad del Software Público, es que el presente trabajo tiene como principal objetivo, indagar en qué aspectos los instrumentos puestos a disposición para el desarrollo de Software Público, contemplan la calidad del software. Específicamente se desarrollará una propuesta inicial que, desde la perspectiva de la calidad de productos software y, tomando como marco de referencia lo establecido en la Familia de Normas ISO/IEC 25000 [8], sirvan como soporte para la identificación y especificación de requisitos de calidad para el desarrollo del Software Público.

En lo sucesivo, en la Sección 2 se presentan los trabajos relacionados con esta propuesta; en la Sección 3 se realiza una descripción de la Familia de Normas ISO/IEC 25000 [8]; en la Sección 4 se detallan y analizan los aspectos principales del Código de Buenas Prácticas para el Desarrollo de Software Público (CBPSP); en la Sección 5 se describen las relaciones y aportes que pueden realizarse al CBPSP desde la Familia de Normas ISO/IEC 25000 [8]; en la Sección 6 se ejemplifica la propuesta, a través de la presentación de un ensayo preliminar; en la Sección 7 se presentan los resultados; por último, en la Sección 8 se exponen las conclusiones y se plantean los trabajos futuros.

2 Trabajos Relacionados

A continuación, se detallan trabajos que están relacionados con los principales aspectos que plantea el desarrollo de la presente propuesta. En este sentido vemos que en [3] se propone un análisis cualitativo de los principales casos de implementación a nivel gubernamental de Software Público en la Argentina, en donde se menciona una serie de condiciones y alternativas para su implementación, como: costos, normativas, tecnología, cultura organizacional, formación del capital humano, entre otras.

En el trabajo denominado Software Público de la Provincia de Buenos Aires [10], se menciona la necesidad de crear un proyecto con una mirada amplia y estratégica que permita una mejor optimización de recursos a la hora de implementar o desarrollar software, específicamente en el ámbito de la Provincia de Buenos Aires.

En [11] [12] [13] y [14] se abordan temas relacionados con la evaluación de la calidad de plataformas y aplicaciones gubernamentales de aspectos relacionados a la accesibilidad, usabilidad e interoperabilidad con distintos enfoques y utilizando estándares propuestos por ISO/IEC.

Visto lo detallado anteriormente podemos ver que existen iniciativas que están orientadas a la mejora del desarrollo del denominado Software Público, tomando en cuenta aspectos relacionados con su implantación y gestión, como así también a lo relacionado a su evaluación con el uso de estándares, particularmente, orientados a aspectos relacionados con la accesibilidad, usabilidad e interoperabilidad.

Existe un tema no menor que está relacionado con las metodologías de desarrollo del software, que, como lo sugiere el propio CBPSP [5] es recomendable el uso de marcos ágiles para el desarrollo de software. En este sentido y en función de la

propuesta que plantea este trabajo se mencionan a continuación algunas iniciativas que abordan la necesidad de incorporar la dimensión de la calidad del software en marcos ágiles para el desarrollo de software.

En [15] se desarrolla una propuesta en la cual, entre otras, se plantea la necesidad de incluir requisitos no funcionales en las historias de usuarios y criterios de aceptación, también y en este sentido en [16] se expresa que Scrum es uno de los marcos ágiles más populares y que la calidad del proceso y del producto final siguen siendo los principales retos de Scrum.

De manera puntual vemos que en un caso sólo se limita a plantear la inclusión de requisitos no funcionales en artefactos puntuales del marco ágil, mientras que en el otro se plantea una modificación en Scrum para que se incorpore un rol específico relacionado a la calidad y se sugiere el uso de un modelo de calidad como el de ISO/IEC 25010 [17].

En los trabajos citados anteriormente, se plantean propuestas para la mejora de la gestión del software público y también se mencionan instrumentos que permiten evaluar dimensiones específicas de la calidad. Estas propuestas que están enfocadas específicamente en el desarrollo del Software Público, no hacen hincapié en la calidad del mismo desde el punto de vista de quien desarrolla el software.

También fue pertinente citar propuestas que abordan la inclusión de aspectos de la calidad del software en los marcos ágiles para el desarrollo de software, ya que estos, y como se mencionó anteriormente son los que sugiere el CBPSP [5] para su adopción; en este sentido, vemos que los trabajos relacionados a esta temática son de gran interés y generan un gran aporte, ya sea, desde el punto de vista técnico como metodológico, pero entendemos que existen aspectos relacionados con la especificación de requisitos de la calidad relacionados al producto software y en particular en el dominio del desarrollo del Software Público que se podrían complementar.

3 Familia de Normas ISO/IEC 25000

El estándar ISO/IEC 25000 [8], conocido también como SQuaRE (del inglés, Systems and software Quality Requirements and Evaluation), es la evolución e integración del modelo de referencia ISO/IEC 9126, y del marco de trabajo ISO/IEC 14598. SQuaRE es una serie organizada lógicamente, enriquecida y unificada, la cual contiene un conjunto de normas, que permiten cubrir dos procesos principales:

1-Especificación de requerimientos de la calidad de software.

2-Evaluación de la calidad del software, apoyados por un proceso de medición de la calidad de software.

El propósito de SQuaRE consiste en ayudar al desarrollo y adquisición del producto de software, a través de la especificación y evaluación de requerimientos de calidad. Establece criterios para la especificación de requerimientos de la calidad del producto de software, su medición y evaluación. Además, incluye un modelo de calidad que permite alinear las definiciones de la calidad del cliente, con atributos del proceso de desarrollo.

Estos modelos de calidad pueden ser utilizados por desarrolladores, adquirientes, evaluadores independientes y personal de aseguramiento y de control de la calidad; es útil particularmente para aquellos quienes son responsables de especificar y evaluar la calidad del producto de software.

En la Norma ISO/IEC 25010 [17] se definen dos tipos de modelos de calidad: calidad del producto y calidad en uso. Si bien se hace mención de ambos modelos, dada las características de este trabajo, solo se desarrollará el modelo de calidad del producto.

El Modelo de calidad del producto está compuesto por 8 características y 32 subcaracterística, las cuales se refieren a las propiedades estáticas del software, y a las propiedades dinámicas del sistema informático. Este modelo es aplicable tanto a sistemas informáticos como a productos de software.

4 Código de buenas prácticas en el desarrollo de software público

El Código de Buenas Prácticas en el Desarrollo de Software Público (CBPSP) [5], es una guía técnica para el desarrollo sustentable de software en la Administración Pública; el mismo propone un marco de referencia para los organismos, el cual provee un conjunto de mejores prácticas para el desarrollo de software público.

La guía técnica para las buenas prácticas de desarrollo de software contiene 7 (siete) puntos principales; estos a su vez, pueden tener algunas sub-guías, las cuales son aplicables a aquellos temas que requieren de mayor profundidad.

A continuación, se mencionan los 7 (siete) apartados que contiene la guía técnica:

- 1) Entender las necesidades de los usuarios
- 2) Aprovechar la nube y la virtualización
- 3) Reutilizar y compartir software y datos
- 4) Probar el servicio de punta a punta
- 5) Proteger al software y a los usuarios
- 6) Utilizar metodologías ágiles e iterativas en todo el ciclo de vida del software
- 7) Documentar y compartir el conocimiento de tus sistemas

Según se detalla en la mencionada guía, estos 7 puntos están divididos en tres niveles de acción:

- El Ideal, es el rumbo aspiracional de mediano-largo plazo, que la ONTI recomienda a las áreas de sistemas en la Administración Pública, en donde se encuentran las mejores prácticas y tendencias en la industria TIC a nivel regional y global.

- El nivel Mínimo, el cual consiste en un punto de transición hacia el Ideal, el cual se pretende que las direcciones de sistemas, logren alcanzar en el mediano plazo a través de buenas prácticas.

- Por dónde empezar: este es el tercer nivel, en el cual se describen los primeros pasos a seguir, para comenzar el camino a la innovación y sustentabilidad tecnológica en el desarrollo de software.

A continuación, y tomando en cuenta el foco del presente trabajo, en esta sección se presenta el análisis realizado de manera específica, para los apartados 1 y 6 del CBPSP [5].

Como se expresó anteriormente, existen tres niveles para cada uno de los apartados; respecto del análisis que deseamos desarrollar, tomaremos el nivel denominado Ideal, dado que este nivel incluye a los otros dos que le preceden.

4.1 Entender las necesidades de los usuarios

Respecto de este apartado, podemos observar que existen cinco ítems, los cuales se detallan de manera resumida a continuación:

- 1) Crear mapas de historias de usuario para entender la experiencia y necesidades de los distintos tipos de usuarios.
- 2) Desarrollar prototipos para probar el servicio con usuarios reales.
- 3) Tomar decisiones basadas en la recolección y análisis de datos objetivos, incluyendo Test A/B.
- 4) Diseñar teniendo en cuenta toda la experiencia del usuario, incluyendo las acciones que realiza fuera de los sistemas digitales.
- 5) Evaluar los servicios con las herramientas que provee la W3C.

De acuerdo al análisis llevado a cabo en este apartado, vemos cuestiones importantes a la hora del inicio del desarrollo de un determinado software, como la de generar historias de usuarios y diseños enfocados en distintos tipos de usuarios, inclusión de test y pruebas de usabilidad, y la sugerencia de evaluar con herramientas como las que provee W3C.

4.2 Utilizar metodologías ágiles e iterativas en todo el ciclo de vida del software

El presente apartado cuenta con diez ítems, los cuales se describen a continuación:

- 1) Construir una visión compartida del producto con el equipo, los interesados, y comprender las necesidades de los usuarios.
- 2) Construir una hoja de ruta a partir de usuarios ideales que respondan a necesidades reales.
- 3) Definir en conjunto con el equipo el objetivo de cada finalización del ciclo de trabajo que oriente.
- 4) Priorizar las entregas de valor, y las tareas necesarias para desplegar tus productos y servicios.
- 5) Desarrollar prototipos y versiones mínimas del producto para validar ideas.
- 6) Planificar las entregas, y reflexionar periódicamente sobre los procesos y los productos.
- 7) Incorporar técnicas de DevSecOps y DevOps, para entregar valor frecuentemente de forma segura.
- 8) Construir un equipo multidisciplinario y lo más autónomo posible para evitar dependencias.
- 9) Desarrollar las capacidades del equipo, utilizando programación por pares.
- 10) Visibilizar las tareas y el flujo de trabajo, para que puedan ser compartidos con tu equipo.

En este apartado, se propone a las metodologías ágiles como marco de trabajo para el desarrollo de software, mencionando la programación por pares, y el uso del marco de trabajo de Scrum. También se hace hincapié en cuestiones relacionadas a mantener

una visión compartida sobre el desarrollo, la entrega frecuente de software y tener siempre presente las necesidades de los distintos tipos de usuario.

4.3 Resumen de los aspectos más relevantes del análisis CBPSP.

Del análisis general llevado a cabo sobre los siete apartados que conforman el CBPSP [5], y en particular el de los apartados 1 y 6, podemos observar que el código constituye una guía genérica que puede ser utilizada por los equipos de desarrollo de software pertenecientes a la Administración Pública.

De manera puntual, podemos ver que se sugiere el uso de metodologías ágiles, como marco para guiar el desarrollo de software, el uso de prácticas que promuevan las entregas continuas de software, tomando en cuenta las necesidades de los distintos tipos de usuarios.

Con referencia al uso de estándares, se mencionan los relacionados con la evaluación de la accesibilidad (herramientas provistas por W3C), y también se mencionan distintos elementos a tener en cuenta relacionados a la interoperabilidad, la usabilidad y la seguridad.

5 Relaciones y Aportes – CBPSP e ISO/IEC 25000

Luego del análisis llevado a cabo en la sección anterior se puede observar que en el CBPSP [5] no se mencionan de manera explícita apartados o referencias que desarrollen aspectos relacionados con la calidad del software, ya sea a nivel de procesos, proyectos, productos o datos.

Un concepto importante asociado a la calidad del software es el relacionado a las propiedades del software [17], donde se describe que las mismas pueden clasificarse en propiedades inherentes y propiedades asignadas.

En la Figura 1, se exhibe la clasificación de las propiedades del software

Propiedades del software	Propiedades inherentes	Propiedades funcionales de dominio específico Propiedades de calidad (adaptación funcional, confiabilidad, eficiencia del desempeño, usabilidad, seguridad, compatibilidad, mantenibilidad, portabilidad)
	Propiedades asignadas	Propiedades de gestión, como, por ejemplo, precio, fecha de entrega, futuro del producto, proveedor del producto.

Figura 1. Propiedades de calidad ISO/IEC 25010 [17]

La clasificación descrita anteriormente, permite profundizar lo analizado en la Sección 4; tomando las propiedades inherentes al software, vemos que desde el CBPSP [5], tanto los aspectos metodológicos propuestos, ya sea como marcos de trabajo para el desarrollo de software, o las mismas pruebas, en su gran mayoría se enfocan en las propiedades funcionales, mientras que existen muy pocas referencias respecto de las propiedades de calidad, por ejemplo, sólo mencionan cuestiones ligadas a la usabilidad, accesibilidad, interoperabilidad y seguridad, que si bien son nombradas, y en algunos casos se recomienda material anexo, se observa poca cobertura hacia estos aspectos.

De acuerdo a lo establecido en el CBPSP [5], en esta sección se llevará a cabo una asociación entre el mencionado Código, y la Familia de Normas internacionales ISO/IEC 25000 [8], las cuáles como se detalló en la sección 3, están enfocadas en la calidad del producto software.

En la Tabla 1 se muestra la asociación entre el CBPSP [5] y las distintas normas pertenecientes a ISO/IEC 25000 [8], las cuales pueden revestir un aporte hacia el mencionado Código.

Tabla 1. Relaciones entre el CBPSP e ISO/IEC 25000 [8]

Apartados CBPSP	Asociación con normas pertenecientes a ISO/IEC 25000
Entender las necesidades de los usuarios: <i>Entender la experiencia y necesidades de los distintos tipos de usuarios.</i> <i>Implementar Test A/B, métricas y pruebas de usabilidad</i>	ISO/IEC 25030 [18] ISO/IEC 25010 [17] ISO/IEC 25023 [19]
Utilizar metodologías ágiles e iterativas en todo el ciclo de vida del software <i>Visión compartida sobre el desarrollo, que existan entregas frecuentes de software y tener siempre presente las necesidades de los distintos tipos de usuario.</i>	ISO/IEC 25030 [18] ISO/IEC 25010 [17] ISO/IEC 25023 [19]

Por otra parte, en la Tabla 2 se detallan los aspectos técnicos y/o metodológicos, que pueden aportar las normas identificadas en la Tabla 1; para ello se toma como ejemplo el primer apartado del CBPSP [5].

Tabla 2. Relaciones entre el CBPSP e ISO/IEC 25000 [8]

CBPSP: <i>Entender la experiencia y necesidades de los distintos tipos de usuarios. Implementar Test A/B, métricas y pruebas de usabilidad.</i>
Aporte desde ISO/IEC 25030 [12]
<ol style="list-style-type: none"> 1. Permite Identificar y caracterizar los de tipos de usuarios que conforman las partes interesadas: <ul style="list-style-type: none"> - Usuarios Principales - Usuarios Secundarios - Usuarios Indirectos 2. Posibilita Identificar las Necesidades de las partes interesadas y derivarlas a necesidades de la Calidad de las partes interesadas. 3. Presenta un Proceso (<i>Proceso de definición</i>) que permite transformar las necesidades de calidad de las partes interesadas en Requisitos de la calidad de las partes interesadas. 4. Presenta un Proceso (<i>Proceso de análisis</i>) que permite transformar los Requisitos de la calidad de las partes interesadas en Requisitos de la calidad del software.
Aporte desde ISO/IEC 25010 [17]
<ol style="list-style-type: none"> 1. De manera puntual esta norma aporta un modelo de calidad de producto software que permite llevar a cabo el <i>proceso de definición</i> planteado en ISO/IEC 25030 [18]. 2. Permite arribar a un modelo detallado que representa los requisitos de calidad de las partes interesadas. Por ejemplo, para las necesidades de las partes interesadas en torno a la Usabilidad se podría definir de esta forma: Característica definida: Usabilidad Subcaracterísticas definidas: - Estética de la interfaz del usuario <ul style="list-style-type: none"> - Facilidad de aprendizaje - Capacidad de reconocer la adecuación
Aporte desde ISO/IEC 25023 [19]

1. Esta norma aporta un modelo de referencia de medición de la calidad del producto software, el cual permite llevar a cabo el *proceso de análisis* planteado por ISO/IEC 25030 [20].

2. Permite arribar a un modelo de medición, compuesto por métricas y función de medición, este modelo representa los requisitos de calidad del software. Por ejemplo y siguiendo con lo planteado para la característica Usabilidad se puede arribar a una serie de medidas asociadas a las diferentes sub-características definidas. Para los fines prácticos solo se detallará una subcaracterística:

Característica definida: Usabilidad

Subcaracterística definidas: - Estética de la interfaz del usuario

Métrica: Estética de apariencia de las interfaces de usuario

Función de medición: $X = A/B$ donde A = Número de interfaces de visualización estéticamente agradables para los usuarios en apariencia y B = Número de interfaces de visualización.

A través de lo desarrollado en esta sección, se pueden evidenciar los aportes que desde la perspectiva de la calidad de productos software, y específicamente la utilización de la familia de normas ISO/IEC 25000 [8], se pueden hacer hacia el CBPSP [5].

Específicamente, podríamos resumir que estos aportes pueden colaborar en la identificación de las necesidades de calidad de las partes interesadas (haciendo uso de la Norma ISO/IEC 25030 [18]), las cuales luego pueden transformarse técnicamente en requisitos de calidad (utilizando la Norma ISO/IEC 25010 [17]), para finalmente arribar a una serie de requisitos de calidad del software, los cuales también son factibles de medir, a través de la Norma ISO/IEC 25023 [19].

6. Ensayo preliminar de la propuesta

En base a lo desarrollado en la sección anterior, se confeccionó y ejecutó un primer ensayo, con el fin de evidenciar los aportes prácticos que pueden realizarse al CBPSP [5], desde la perspectiva de la calidad de producto software, y por medio del uso de la Familia de Normas ISO/IEC 25000 [8].

Para la realización de este ensayo, se tomó como referencia una Facultad de Ingeniería perteneciente a una Universidad Pública, la cual cuenta con un área de desarrollo de software.

De manera puntual, escogimos una Web App que fue desarrollada por el organismo antes mencionado; el principal objetivo para la cual fue desarrollada esta Web App, es que los estudiantes puedan realizar distintos tipos de gestiones académicas.

De manera general, esta área de desarrollo utilizó como guía lo establecido en el CBPSP [5]; por otra parte, cabe mencionar que el desarrollo estuvo guiado por el marco de trabajo propuesto por Scrum.

En la Tabla 3 se muestran los ítems priorizados de la Pila del Producto.

Tabla 3. Pila del Producto de la Web App

Id Ítem	Ítem Pila del Producto
01	Inicio de sesión.
02	Visualización del estado académico.
03	Inscripción al cursado de materias.
04	Visualización del cursado actual.
05	Visualización de días de cursado.
06	Visualización de datos de cursado por ciclo lectivo.

07	Visualización de exámenes por ciclo lectivo.
08	Visualización de inasistencias de registradas en ciclo lectivo.
09	Visualización de correlatividades a cursar.
10	Visualización de Materias del Plan.
11	Solicitud de Certificados.
12	Inscripción a exámenes finales
13	Visualización de datos de inscripción a exámenes.
14	Visualización de correlatividades para rendir.
15	Solicitud de certificado de postergación de examen.

Antes de desplegar la Web App en ambiente productivo, y con el fin de incorporar a las historias de usuarios aspectos relacionados a la calidad, se seleccionó un grupo de estudiantes con el propósito de llevar a cabo Pruebas de Aceptación de Usuario.

Con el objetivo de determinar las necesidades de calidad de las partes interesadas, las pruebas de aceptación fueron guiadas según el esquema planteado por ISO/IEC 25030 [18]. A continuación, en la Tabla 4 se expresan las necesidades de calidad, especificadas en formato de Historias de Usuario (HU).

Por cuestión de espacio en este artículo, se muestran los datos relacionados a los “usuarios principales”, para la funcionalidad “Inscripción a exámenes finales”.

Tabla 4. Pila del Producto de la Web App

Usuario: Estudiante	
Tipo de Usuario: Principal	
HU-11: Inscripción a exámenes finales	
Descripción: <i>Como</i> estudiante <i>Quiero</i> inscribirme al examen final de una materia <i>Para</i> poder rendirla en una fecha de examen determinada	
Necesidades de calidad de las partes interesadas (Usuarios principales)	
Id	Necesidades de Calidad
1	Que la inscripción se pueda realizar de manera más simple.
2	Que pueda ser realizada por personas que cuentan con alguna discapacidad.
3	Que la interfaz sea más agradable e intuitiva.
4	Que esté disponible en los días en que se realizan inscripción a exámenes finales.
5	Que sea compatible con todos dispositivos móviles, por ejemplo, no funciona en iPhone.
6	Que la interfaz se puede adaptar en todos los dispositivos móviles, por ejemplo, en algunos no es 100% responsivo.

Una vez identificadas las necesidades de calidad de los usuarios, aplicamos el proceso de definición establecido en ISO/IEC 25030 [18], haciendo uso del modelo de calidad de producto software, propuesto en la Norma ISO/IEC 25010 [17]. Como resultado, logramos determinar los requisitos de la calidad de las partes interesadas, a través de la identificación de características y sub-características del modelo de calidad, que están asociadas a las necesidades de calidad antes detectadas. En la Tabla 5 se muestra el resultado de lo antedicho.

Por cuestiones de espacio, solo se transcribe el Id de las necesidades de calidad (Id_NC).

Tabla 5. HU - Requisitos de calidad de las partes interesadas

Usuario: Estudiante

Tipo de Usuario: Principal		
HU-11: Inscripción a exámenes finales		
Descripción: <i>Como</i> estudiante <i>Quiero</i> inscribirme al examen final de una materia <i>Para</i> poder rendirla en una fecha de examen determinada		
Requisitos de la calidad de las partes interesadas (Usuarios principales)		
Id_NC	Característica	Subcaracterística
1	Usabilidad	Facilidad de Aprendizaje
2	Usabilidad	Accesibilidad
3	Usabilidad	Estética
4	Fiabilidad	Disponibilidad
5	Compatibilidad	Coexistencia
6	Portabilidad	Adaptabilidad

Luego de la identificación de los requisitos de calidad de las partes interesadas, se aplica el proceso de análisis, donde se utiliza la Norma ISO/IEC 25023 [19]; como resultado de esta acción, podremos establecer las medidas para cada aspecto de calidad que fue identificado a través de una sub-característica de calidad, pudiendo de esta manera obtener los requisitos de calidad del software. En la Tabla 6 se muestra cómo quedarían especificados estos requisitos.

Tabla 6. HU - Requisitos de calidad del software

Usuario: Estudiante			
Tipo de Usuario: Principal			
HU-11: Inscripción a exámenes finales			
Descripción: <i>Como</i> estudiante <i>Quiero</i> inscribirme al examen final de una materia <i>Para</i> poder rendirla en una fecha de examen determinada			
Requisitos de calidad del software (Usuarios principales)			
Id_NC	Característica	Subcaracterística	Medida de Calidad
1	Usabilidad	Facilidad de Aprendizaje	Interfaz de usuario auto explicativa
2	Usabilidad	Accesibilidad	Accesibilidad para usuarios con discapacidad
3	Usabilidad	Estética	Estética de apariencia de las interfaces de usuario
4	Fiabilidad	Disponibilidad	Disponibilidad del sistema
5	Compatibilidad	Coexistencia	Coexistencia con otros productos
6	Portabilidad	Adaptabilidad	Adaptabilidad del entorno operativo

Luego de explicitados los requisitos de calidad del software, para una funcionalidad específica, se procede a la definición de los criterios de aceptación de calidad de producto (CA-CP). En la Tabla 7 se muestra cómo quedarían redactados cada uno de estos criterios de aceptación para la HU en cuestión.

Tabla 7. HU - Requisitos de calidad

Usuario: Estudiante
Tipo de Usuario: Principal
HU-11: Inscripción a exámenes finales
Descripción: <i>Como</i> estudiante

Quiero inscribirme al examen final de una materia	
Para poder rendirla en una fecha de examen determinada	
Criterios de Aceptación para los Requisitos de calidad del producto	
Id NC	CA-RCP
1	Debe simplificarse la interfaz de usuario con pasos más sencillos, para que el proceso de inscripción resulte más fácil y ágil.
2	Los estudiantes que posean alguna discapacidad deben poder acceder a inscribirse a examen.
3	Debe estar implementado un cambio estético en la interfaz del usuario, de manera que pueda verse la diferencia con la estética actual.
4	Debe asegurarse la disponibilidad de la funcionalidad, sobre todo en días de inscripción a examen.
5	El proceso de inscripción debe poderse realizar desde cualquier dispositivo móvil, que sea compatible con los diferentes fabricantes (iPhone, Samsung, etc.).
6	Los elementos o componentes en la inscripción deben ser 100% responsivos, para evitar una mala experiencia de interacción de usuario.

Tomando como base los CA-CP definidos, el equipo de desarrolladores podrá elaborar un plan de pruebas, para esa fase, y si bien no es objeto de este trabajo, también se pueden incorporar aspectos relacionados a la medición, asociando a las medidas de calidad que fueron identificadas, funciones de medición que provee la Norma ISO/IEC 25023 [19].

7. Resultados

Luego de haber analizado el Código de Buenas Prácticas para el Desarrollo de Software (CBPSP), se observa que el mismo representa una guía general para aquellas reparticiones pertenecientes a la Administración Pública, que cuenten con equipos y/o áreas de desarrollo y mantenimiento de software.

Si bien se llevó a cabo un análisis completo del CBPSP [5], dado el objetivo planteado en este trabajo, el estudio se focalizó en dos apartados específicos. Según se detalla en la Sección 4, el uso de metodologías ágiles como marco para guiar el desarrollo de software, las prácticas que promuevan las entregas continuas de software, y poner las necesidades de los usuarios en el centro de la escena, fueron los aspectos más relevantes.

De acuerdo a lo desarrollado en la Sección 5, se observan las asociaciones y aportes que desde las normas ISO/IEC 25030 [18], ISO/IEC 25010 [17] e ISO/IEC 25023 [19] se pueden realizar hacia el CBPSP [5] en los aspectos relacionados a la calidad del software, de manera específica en lo que respecta a los marcos que faciliten la identificación de los distintos tipos de usuarios y la especificación de requisito de calidad del software a través del modelo de calidad propuesto.

Al ensayar esta propuesta en un caso real, pudimos observar que al haber implementado lo propuesto en este trabajo fueron identificados los aspectos relacionados a la calidad, los cuales fueron detectados y especificados. Por ejemplo, se identificaron 4 características, 6 subcaracterísticas, y 6 medidas de calidad asociadas a una funcionalidad específica del producto. Las cuatro principales características de calidad que fueron detectadas y especificadas fueron: Usabilidad, Fiabilidad, Compatibilidad y Portabilidad.

La identificación y especificación de los factores de calidad, antes mencionados, permitió llevar a cabo la redacción de 6 criterios de aceptación, para la funcionalidad analizada, con una marcada especificidad en lo que respecta a calidad de producto software, lo que derivará luego en la inclusión de estos aspectos en los planes de pruebas correspondientes. Asociado a esta última parte, el equipo de desarrollo podrá contar, no solo, con las medidas de calidad, sino también, con las funciones de medición correspondiente, las cuales se proveen en la norma ISO/IEC 25023 [19].

7. Conclusiones y Trabajos Futuros

Como primer punto, es importante destacar las iniciativas que desde el Estado se llevan adelante, en pos de fomentar y mejorar el desarrollo del software en la Administración Pública, poniendo a disposición recursos técnicos y metodológicos para que ello sea posible.

Del análisis llevado a cabo del CBPSP [5], se pudieron detectar las relaciones existentes entre los apartados específicos del código, y las normas pertenecientes a ISO/IEC 25000 [8]. Las normas específicas detectadas realizan aportes en los aspectos relacionados a identificar a los usuarios pertenecientes a las partes interesadas, detectando en primera instancia sus necesidades de calidad, para luego poder explicitarlas de manera cuantitativa y cualitativamente, a través del modelo de calidad y las medidas de calidad.

Lo desarrollado en las secciones 4 y 5, permitió establecer un marco con el cual se pudo ejecutar un primer ensayo (Sección 6). Si bien los resultados son preliminares, muestran de manera concreta los aportes y beneficios de poder contar con un marco, que permita favorecer la especificación de requisitos de calidad de productos software.

Según lo expresado anteriormente, el desarrollo del software en ámbito de la Administración Pública, es heterogéneo. Su evolución dependerá en gran medida de los recursos a los que pueda acceder cada repartición. Estas iniciativas, como tantas otras, intentan fortalecer los instrumentos que el estado pone a disposición para el desarrollo del Software Público.

Este trabajo también deja en evidencia la necesidad de profundizar y desarrollar propuestas futuras, como, por ejemplo, el desarrollo de nuevos ensayos o casos que permitan detectar las fortalezas y debilidades de esta propuesta en distintos ámbitos de desarrollo; elaborar una guía que permita conducir el proceso de especificación de requisitos de calidad de producto software; establecer criterios para la elaboración de planes de pruebas orientados a factores de calidad de producto.

Referencias

- [1] M. Pliattini, F. García, I. García y F. Pino, Calidad de Sistemas de Información-4º Edición Ampliada y Actuaizada, Madrid: RA-MA, 2018.
- [2] Ministerio de Modernización, «Decreto 434/2016 Plan de Modernización del Estado de la República Argentina.» 2016. [En línea]. Available: https://www.argentina.gob.ar/sites/default/files/cofemod_documentos_plan_de_modernizacion_del_estado_2016.pdf. [Último acceso: 2019].

- [3] G. Fantin, F. Salemo y L. Ordinez, «Prospectiva para la Implementación de Software,» *JAIIO - SIE 2016 10º Simposio de Informática en el Estado*, 2016.
- [4] OFICINA NACIONAL DE TECNOLOGÍAS DE INFORMACIÓN, «<https://www.boletinoficial.gob.ar>,» 29 08 2019. [En línea]. Available: <https://www.boletinoficial.gob.ar/detalleAviso/primera/215325/20190902?busqueda=2>. [Último acceso: 2019].
- [5] ONTI, «Código de Buenas prácticas en el desarrollo de software público,» ONTI, 2019. [En línea]. Available: <https://www.argentina.gob.ar/onti/codigo-de-buenas-practicas-para-el-desarrollo-de-software-publico>. [Último acceso: 2021].
- [6] J. GARZÁS, M. RODRÍGUEZ, D. GAITERO, P. MINGO, D. GARCÍA, J. CORROCHANO, E. SÁNCHEZ y A. OTERINO, I Jornadas sobre Calidad del Producto Software e ISO 25000, Santiago de Compostella: 233 Grados de TI S.L., 2014.
- [7] R. Pressman, *Ingeniería de Software.*, 6 ed., McGraw-Hill, 2005.
- [8] ISO/IEC 25000:2014, *Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)*, ISO-Guide to SQuaRE, 2014.
- [9] A. Villalta y J. Carvallo, «Modelos de calidad de software: Una revisión sistemática de la literatura,» de *CEDIA*, Maskana, 2015.
- [10] G. Maldonado, «Software Público de la Provincia de Buenos Aires,» *JAIIO - Jornadas Argentinas de Informática*, 2021.
- [11] M. Valderrey y P. Thomas, «Interoperabilidad: Acceso a productos y servicios utilizando estándares,» *JAIIO - SIE*, 2019.
- [12] M. Castro, V. Rivero, L. Vargas, J. Farfán, E. Reinoso y M. Aparicio, «Evaluando la Calidad de Uso en aplicaciones web de gobierno electrónico y ciencia ciudadana con estándares ISO/IEC,» *CACIC - Congreso Argentino de Ciencias de la Computación*, 2019.
- [13] A. Carrizo, G. Gaetán, A. Martín y V. Saldaña, «Aplicaciones de Gobierno Móvil en Argentina-Un Estudio de Usabilidad,» *JAIIO - SIE*, 2019.
- [14] M. Castro, V. Rivero, L. Vargas, J. Farfán, E. Reinoso y M. Aparicio, «Calidad de Uso en un Sitio de Gobierno Electrónico: Evaluación con ISO 25062,» *JAIIO - SIE*, 2019.
- [15] M. Estayno y J. Meles, «El Rol del Product Owner en la definición y validación de las user stories,» *Ciencia y Tecnología*, n° 14, pp. 145-162, 2014.
- [16] C. Tona, R. Juárez, S. Jiménez, A. Quezada, C. Guerra y R. González, «Scrumlity: An Agile Framework Based on Quality Assurance,» de *9th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 2021.
- [17] ISO/IEC 25010:2011, *Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-System and software quality models.*, ISO, 2011.
- [18] ISO/IEC 25030:2007, *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements.*, ISO, 2007.
- [19] ISO/IEC 25023:2016, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality.*, ISO, 2016 (E).
- [20] IRAM ISO/IEC 25030:2019, *Ingeniería de Software – Requisitos de la calidad del producto de software (SQuaRE) – Requisitos de Calidad.*, Buenos Aires: IRAM, 2019.