

Evaluación de la calidad en aplicaciones móviles

Susana I. Herrera¹, Pablo J. Najjar¹, Rosa Palavecino¹ & Jorge L. Goñi²

(1) *Instituto de Investigación en Informática y Sistemas de Información, Facultad de Ciencias Exactas y Tecnologías, Universidad Nacional de Santiago del Estero.*

{sherrera, rosypgg}@unse.edu.ar, najarpablo@yahoo.com.ar

(2) *Facultad de Humanidades, Ciencias Sociales y de la Salud, Universidad Nacional de Santiago del Estero.*

jlgoni@live.com.ar

RESUMEN: La Calidad es una característica del software o de los sistemas que no es factible de ser medida. La calidad se evalúa o se calcula, a partir de la medición de ciertos atributos relacionados con ella. Si bien existen una gran cantidad de estudios sobre la calidad de las aplicaciones en general, no ocurre lo mismo con la calidad de las aplicaciones móviles. Es por ello que este trabajo pretende contribuir a la evaluación de la calidad de las aplicaciones móviles, desde el enfoque de la calidad del producto, abordando una de sus principales características: la eficiencia. Para ello, se toma como referencia la Norma ISO 25000 y el trabajo académico y de investigación desarrollado por el grupo de investigación GIDIS WEB de la Universidad Nacional de La Pampa. En base al método GOCAME, se definen las subcaracterísticas de la eficiencia, sus atributos, sus métricas e indicadores. Se aplica sobre prototipos de aplicaciones móviles para turismo desarrollados con dos arquitecturas alternativas: web e híbrida.

1. INTRODUCCIÓN

Este trabajo se desarrolla en el marco del proyecto de investigación “Optimización de la calidad de los Sistemas Móviles mediante la implementación de nuevas arquitecturas, realidad aumentada, técnicas de visualización y redes móviles Ad-Hoc”. Cuyo objetivo es la optimización de las aplicaciones móviles utilizando las tecnologías mencionadas. El equipo de investigación está conformado por docentes y alumnos avanzados del Departamento de Informática de la Facultad de Ciencias Exactas y de la Facultad de Humanidades de la Universidad Nacional de Santiago del Estero (UNSE), del Departamento de Sistemas de la Facultad de Tecnologías y Ciencias Aplicadas de la Universidad Nacional de Catamarca (UNCa) y del Departamento de Computación de la Facultad de Ciencias Exactas de la Universidad Nacional de Salta (UNSa).

La calidad es una característica del software o de los sistemas que no es factible de ser medida. La calidad se evalúa o se calcula, a partir de la medición de ciertos atributos relacionados con la calidad. Un atributo es una propiedad física o abstracta que se puede medir usando una métrica. Esos atributos definen el valor de un concepto calculable el cual es evaluado, junto a otros, para evaluar la calidad.

La calidad puede ser vista desde diferentes puntos de vista: del desarrollador, del usuario, etc. Estos puntos de vista están relacionados con los estudios de calidad interna, externa o en uso. Lo importante es que se desea comprender y evaluar la calidad de un producto software para poder tomar decisiones respecto al mismo.

Primero se debe medir atributos, luego evaluar la calidad, así se logra comprender la calidad para poder hacer recomendaciones sobre un producto. A raíz de las recomendaciones se producen cambios y, por consiguiente, mejoras en el sistema.

Si bien existen una gran cantidad de estudios sobre la calidad de las aplicaciones en general, no ocurre lo mismo con la calidad de las aplicaciones móviles. Es por ello que este trabajo pretende contribuir al análisis de la calidad de las aplicaciones móviles, abordando una de las principales características de la calidad: la eficiencia. Se trabaja con las aplicaciones móviles para turismo desarrolladas en el proyecto de investigación mencionado (Herrera, Najjar, Ledesma & Rocabado, 2012).

Se toma como referencia de calidad la Norma ISO 25000 (ISO, 2011) y el trabajo académico y de investigación desarrollado por el grupo de investigación GIDIS WEB de la Universidad Nacional de La Pampa en Argentina (Becker, Lew & Olsina, 2012; Olsina, Lew, Dieser & Rivera, 2012; Olsina, Lew, Dieser & Rivera, 2011;

Olsina & Rossy, 2002; Lew, Olsina, Beck & Zhang; 2011).

A continuación se presentan los principales aspectos de la calidad y luego se particulariza la característica *eficiencia* en aplicaciones móviles.

2. MARCO TEORICO

2.1. La calidad según el estándar ISO/IEC 25000

Las ISO 25000 (ISO, 2011) constituyen una familia de normas que sustituyen y amplían las normas ISO 9126 (Calidad de un Producto Software) y la ISO 14598 (Evaluación del Software) desarrollada por el subcomité SC 7 (Ingeniería de software y sistemas) del Comité Técnico Conjunto ISO/IEC JTC 1. Es conocida con el nombre de SQuaRE (Requisitos y Evaluación de Calidad de Productos de Software) y la misma se organiza en cinco capítulos.

En concordancia con la filosofía de los modelos clásicos, el modelo de referencia para la medición de la calidad del producto software de la norma ISO/IEC 25000 establece que la calidad del producto software está determinada por características que, a su vez, están determinadas por subcaracterísticas. Estas pueden ser valoradas a través de ciertos atributos o aspectos que son medidos con ciertas métricas y herramientas.

La norma ISO/IEC 25010 contempla dos modelos de calidad:

- Modelo de Calidad de Producto (Vista Interna / Externa): categoriza las propiedades de la calidad del producto o sistema en ocho características: funcionalidad, eficiencia, compatibilidad, usabilidad, confiabilidad, seguridad, capacidad de mantenimiento y portabilidad. Cada característica está compuesta de un conjunto de subcaracterísticas. El modelo de calidad de producto puede ser aplicado a un producto software o a un sistema computacional que incluye software, dado que la mayoría de las subcaracterísticas son relevantes tanto para el software como para los sistemas.
- Modelo de Calidad en Uso (Vista en Uso): define cinco características referidas a las salidas de la interacción con el sistema: efectividad, eficiencia, satisfacción, libre de riesgos, contexto. Cada característica puede ser asignada a diferentes actividades e los stakeholders, por ejemplo, la interacción de un operador o el mantenimiento de un desarrollador. La calidad en uso de un sistema caracteriza el impacto que un producto (sistema o software) tiene en los stakeholders. Está determinado por la calidad del ambiente del software, hardware y operativo, y por el ambiente de los usuarios,

tarefas y social. Todos estos factores contribuyen a la calidad en uso del sistema.

Además, la norma ISO/IEC 25012 contempla el Modelo de Calidad de Datos.

En las tablas 1 y 2 se presentan las características y subcaracterísticas de los modelos de calidad de uso y de calidad del producto.

Tabla 1. Modelo de calidad en uso.

Características	Sub-Características
Efectividad	• Efectividad
Eficiencia	• Eficiencia
Satisfacción	• Utilidad • Confiianza • Placer • Confort
Libre de Riesgo	• Mitigación del riesgo económico • Mitigación del riesgo en cuanto seguridad y salud • Mitigación de riesgos ambientales
Contextualidad	• Completitud Contextual • Flexibilidad

La calidad del producto y la calidad en uso están muy relacionadas, es por ello que muchos factores que inciden en una de ellas, repercuten en la otra. Por ejemplo: un software con alta funcionalidad y complejidad, ejecutado sobre una máquina con bajas prestaciones, provocará una baja eficiencia en el usuario final, independientemente del factor humano. Esto es muy común en entornos móviles donde los recursos principales del dispositivo donde se ejecutan las aplicaciones son escasos.

Tabla 2. Modelo de calidad de producto.

Características	Sub-Características
Funcionalidad	• Completitud • Correctitud • Oportunidad
Fiabilidad	• Madurez • Tolerancia a fallos • Disponibilidad • Capacidad de Recuperación
Usabilidad	• Capacidad de ser reconocido apropiadamente • Facilidad de aprendizaje • Operabilidad • Interfaz de usuario (estética) • Protección contra errores del usuario • Accesibilidad
Eficiencia	• Comportamiento en el tiempo • Utilización de los Recursos • Capacidad

Tabla 2. Modelo de calidad de producto (continuación).

Características	Sub-Características
Mantenibilidad	<ul style="list-style-type: none"> • Modularidad • Reusabilidad • Capacidad de análisis • Capacidad de ser modificado • Capacidad de ser probado
Portabilidad	<ul style="list-style-type: none"> • Adaptabilidad • Facilidad de instalación • Facilidad de ser reemplazado
Compatibilidad	<ul style="list-style-type: none"> • Coexistencia • Interoperabilidad
Seguridad	<ul style="list-style-type: none"> • Confidencialidad • Integridad • No-Repudio • Responsabilidad • Autenticidad

Las medidas de calidad software (Software Quality Measures) indican las características y subcaracterísticas de calidad del producto software. El valor de estas medidas de calidad software se obtiene por la aplicación de una función de medición (Measurement Function) a los elementos de medición de la calidad (Quality Measure Elements). Los elementos de medición de la calidad son atributos o medidas base o medidas derivadas obtenidas según describe el método de medición correspondiente (Measurement Method), de acuerdo a la ISO/IEC 15939. Cabe resaltar que las normas ISO/IEC 9126 y 25000 establecen cuáles son las características de la calidad de un producto software y sus subcaracterísticas, pero no indican qué medidas de calidad indican una característica o subcaracterística.

2.2. Evaluación de la calidad según GIDIS WEB

En Argentina, el grupo de investigación GIDIS WEB, ha venido trabajando en métodos y procesos de medición y evaluación de la calidad de los sistemas desde hace varios años atrás. En los últimos tiempos se acentuó el estudio de la calidad de las nuevas generaciones de aplicaciones web y de aplicaciones móviles.

Uno de los primeros pasos para realizar la evaluación consiste en definir los requisitos no funcionales, lo cual se realiza generalmente a través de modelos de calidad y también a través de un marco de calidad que define relaciones entre los modelos. Los modelos de calidad pueden estar enfocados a diferentes categorías de entidades: producto, sistema, sistema en uso. Los productos son entidades que se encuentran en las primeras etapas del ciclo de vida del software; los sistemas de información son productos ejecutables en un contexto específico, que pueden incluir tanto hardware como software, y el sistema de información en uso son los sistemas

anteriores pero operados por usuarios reales en contextos reales de uso (ver Figura 1).

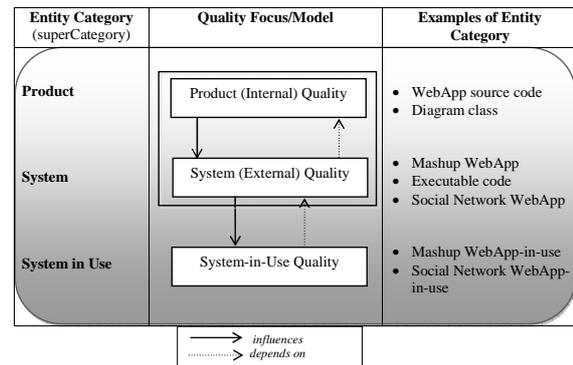


Figura 1. Entidades y Modelos de Calidad (Olsina et al., 2012).

Entre las producciones más importantes del grupo GIDIS WEB se encuentran la elaboración de un marco para la medición y evaluación de la calidad -C-INCAMI- y, además, un modelo para el proceso de evaluación -GOCAME-. A continuación se detallan los mismos.

El marco C-INCAMI, marco para la medición y evaluación de la calidad basado en una ontología de métricas e indicadores INCAMI: Information Need, Concept model, Attribute, Metric and Indicator (Olsina, Pappa, Molina, 2008); define la base conceptual para el dominio de medición y evaluación (M&E). Diseñado para satisfacer una necesidad de información específica en un contexto dado definiendo todos los conceptos y relaciones que se usan en todas las actividades de M&E. El marco tiene seis componentes: definición del proyecto, especificación de requisitos no funcionales, especificación de contexto, diseño e implementación de la medición, diseño e implementación de la evaluación, especificaciones de análisis y recomendación. En la figura 2 se presentan las ontologías correspondientes a dos de dichos componentes: requisitos no funcionales y contexto. Los componentes NFRS especifican la necesidad de información de cualquier proyecto M&E; es decir, el propósito (comprender, predecir, mejorar, etc.) y el punto de vista del usuario (desarrollador, usuario, etc.). Se enfoca en un concepto calculable y especifica la categoría de entidad a evaluar (por ejemplo, un sistema, un sistema en uso), por medio de una entidad concreta (por ejemplo, cúspide.com). Por otro lado, el concepto calculable seleccionado y sus subconceptos pueden ser representados por un modelo de concepto donde las hojas de un modelo instanciado son atributos asociados con una categoría de entidad. De hecho, una entidad concreta pertenece a una categoría de entidad.

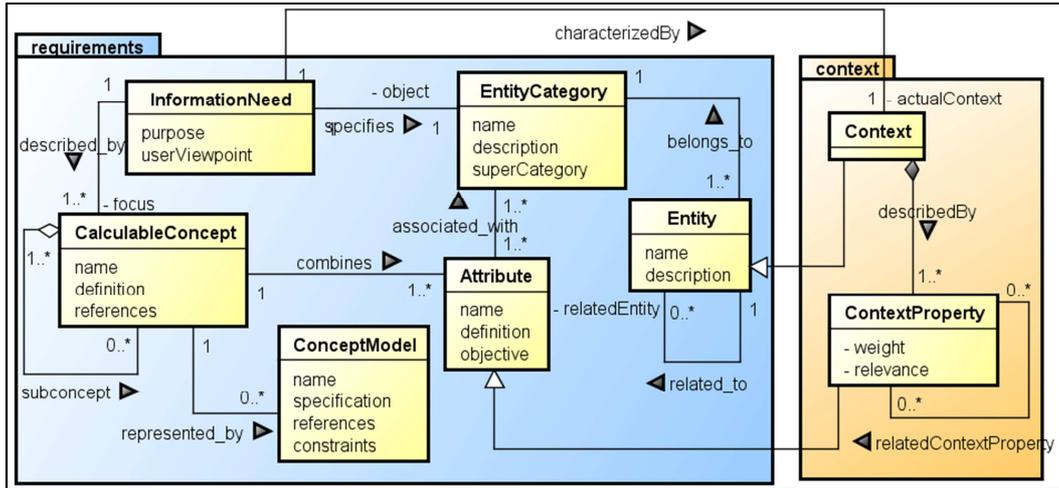


Figura 2. Componentes Requisitos no Funcionales y de Contexto del C-INCAMI (Olsina et al., 2011).

Por otra parte, la estrategia GOCAME, Estrategia integrada de medición y evaluación de la calidad, Goal-Oriented Context-Aware Measurement and Evaluation (Lew et al., 2011), propone un método integral que involucra tres pilares importantes para la medición y evaluación de la calidad de un software: una base conceptual, un proceso y un método. En cuanto a la base conceptual, se

propone una ontología que incorpora conceptos y sus relaciones referidos a: requisitos no funcionales (a los cuales hace referencia la calidad), contexto, medición, evaluación, proyecto. El proceso hace referencia al qué se debe medir y el método hace referencia a cómo medir.

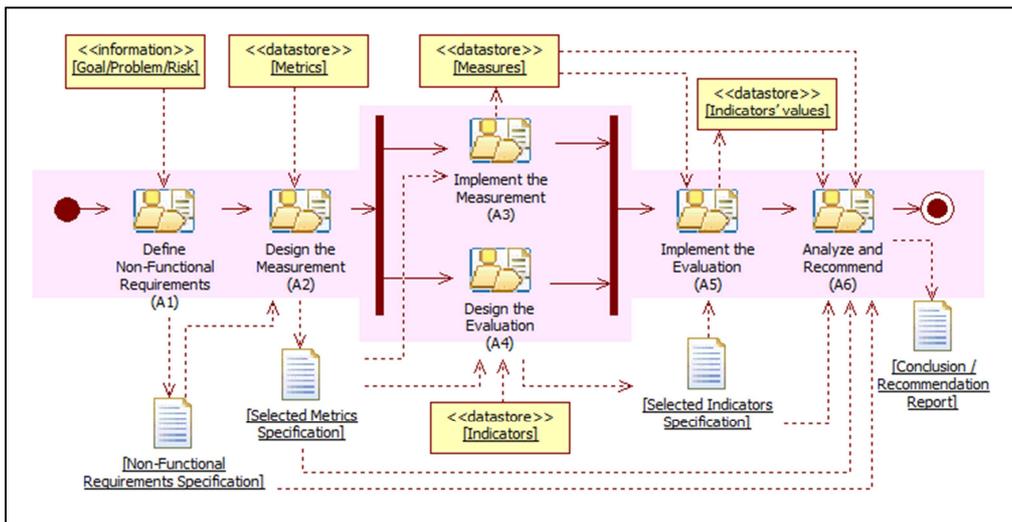


Figura 3. Estrategia GOCAME para la medición y evaluación de la calidad (Lew et al., 2011).

Teniendo en cuenta el marco conceptual presentado y el estándar ISO sobre los procesos de medición y evaluación, el proceso de la estrategia GOCAME propone las siguientes actividades fundamentales (ver figura 3):

1. Definición de los requisitos no funcionales o NFR
2. Diseño de las mediciones
3. Diseño de la evaluación
4. Implementación de las mediciones
5. Implementación de la evaluación
6. Análisis y recomendación

El proceso sigue un enfoque orientado a objetivos. La actividad 1, definición de requisitos no funcionales, tiene un objetivo principal como entrada y un documento de especificación no funcional como salida. Luego, la actividad de diseño de mediciones permite identificar las métricas que se usarán para cuantificar los atributos -las métricas se buscan y seleccionan de un repositorio o catálogo de métricas-; la salida es un documento de especificación de métricas. Una vez que esto está realizado, se pueden llevar a cabo las actividades de diseño de la evaluación e implementación de las mediciones. La actividad

de diseño de la evaluación permite identificar los indicadores elementales y globales que se van a usar. La actividad de implementación de mediciones usa las métricas especificadas para obtener las mediciones, las cuales se guardan en un repositorio de mediciones. Luego, se puede realizar la actividad de implementación de la evaluación. Finalmente, la actividad de análisis y recomendación tiene como entrada los valores de medición e indicadores, el documento de especificación de requerimientos y las especificaciones de las métricas e indicadores.

En cuanto a los métodos y herramientas, la estrategia GOCAME propone la metodología Web-QEM, Web Quality Evaluation Method (Olsina et al., 2002). Mientras las actividades determinan “qué” es lo que se debe hacer, los métodos describen “cómo” realizar esas actividades, las cuales pueden estar automatizadas mediante herramientas.

2.3. La eficiencia como característica de la calidad

En esta propuesta se trabaja en el marco del Modelo de Calidad de Producto. Desde este punto de vista, se define a la eficiencia como la capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas (Piattini, García & Caballero, 2007).

Cabe resaltar que la eficiencia en la calidad del producto incide en el Modelo de Calidad en Uso, ya que influye en que los usuarios puedan alcanzar objetivos con efectividad en un contexto determinado; en un entorno móvil, esto se da con mayor frecuencia. Es por ello que se considera que poniendo énfasis en la etapa de desarrollo del producto software, se podrá garantizar un buen desempeño del mismo y así mejorar su calidad en uso.

Los recursos considerados para definir la eficiencia del producto pueden ser: otros productos software, la configuración del software y del hardware del sistema, y materiales como impresoras, o unidades de almacenamientos.

La eficiencia se determina en función de tres subcaracterísticas:

- Comportamiento respecto al tiempo: Grado en el cual un sistema en ejecución cumple sus requisitos de tiempos de procesamiento y de respuesta y tasas de throughput (rendimiento, caudal de datos).
- Utilización de los Recursos: Grado en el cual la cantidad y tipos de recursos usados por un sistema en ejecución cumplen sus requerimientos. Se incluyen los recursos humanos.
- Capacidad: Grado en el cual los límites máximos de los parámetros del producto

software o sistema cumplen con los requerimientos. Los parámetros pueden incluir el número de artículos que pueden ser almacenados, el número de usuarios simultáneos, el ancho de banda de la comunicación, el rendimiento de las transacciones y el tamaño de la base de datos.

3. EVALUACION DE LA EFICIENCIA EN APLICACIONES MÓVILES

3.1. La eficiencia en aplicaciones móviles

Como está visto, la eficiencia en aplicaciones en general está ligada directamente a la cantidad y forma en que la aplicación consume o utiliza los recursos para poder cumplir con exactitud los requerimientos impuestos.

En la actualidad, las aplicaciones convencionales se ejecutan en un entorno controlado con equipos de computación con grandes prestaciones y bajo una red fija que, con la tecnología, actual transfieren datos a gran velocidad.

Sin embargo, en un entorno móvil, donde el usuario necesita estar siempre “conectado”, no ocurre lo mismo. Este entorno impone limitaciones de recursos a las aplicaciones móviles. Por ejemplo: El consumo de energía de los recursos (cpu, pantalla, wifi, 3G, GPS) tiene un impacto directo sobre la duración de la batería del dispositivo. La frecuencia de la comunicación del dispositivo móvil con un servidor debe encontrar un equilibrio entre la obtención de datos y una buena experiencia de usuario sin que se agote la batería. Un uso desmedido del ancho de banda provoca retrasos que afectan el normal funcionamiento de la aplicación y la experiencia del usuario. Tal es la criticidad de esta situación, que los sistemas operativos Android y Windows Phone prevén la suspensión repentina de las aplicaciones en el ciclo de vida de las mismas para una mejor redistribución de los recursos.

Los recursos críticos que deben contemplarse son: el ancho de banda, CPU, batería, memoria principal, Wifi, 3G.

El objetivo de esta propuesta es determinar la eficiencia de aplicaciones móviles con diferentes arquitecturas y para ello es necesario evaluar previamente las subcaracterísticas: tiempo, recursos y capacidad. Estas subcaracterísticas implica la necesidad de medir aspectos como el ancho de banda, CPU, batería, memoria principal, Wifi, 3G. Algunas de las herramientas que permiten medir los mencionados aspectos y que fueron seleccionadas para ser utilizados en esta propuesta son las siguientes: PowerTutor y Traffic Monitor.

Se evalúa la eficiencia de dos prototipos de aplicaciones de m-turismo, los cuales se desarrollaron con diferentes arquitecturas

(Herrera et al., 2012): híbrida y web. Dichas arquitecturas se evalúan en dos ambientes, Mobile y WI-FI, a partir de los mismos atributos de calidad. Para evaluar la calidad, se siguió un procedimiento basado en el método GOCAME.

Los pasos seguidos son:

1. Desarrollo de los prototipos de aplicaciones con arquitectura híbrida y web
2. Definición de los casos de prueba y diseño y preparación de los escenarios de prueba
3. Evaluación de la calidad de los prototipos
 - a) Definición de los requisitos no funcionales o NFR
 - b) Diseño de las mediciones
 - c) Diseño de la evaluación
 - d) Implementación de las mediciones
 - e) Implementación de la evaluación
4. Análisis y recomendación

A continuación se presentan los resultados de las etapas 1 a 3.c, las cuales responden a las tareas de diseño de las pruebas y mediciones.

3.2 Desarrollo de prototipos de m-turismo

Dada la necesidad de contar en la región con aplicaciones de turismo basadas en posicionamiento, se desarrollaron prototipos en este dominio de aplicación.

Si bien son dos prototipos, uno con arquitectura cliente-servidor híbrida y otro con arquitectura cliente-servidor web, ambos responden al mismo conjunto de requisitos. Fueron desarrollados siguiendo el modelo de desarrollo ágil Programación Extrema o XP.

En cuanto a la funcionalidad, ambos prototipos responden a la siguiente *historia de usuario*:

Un turista visita por primera vez la ciudad de Termas de Río Hondo. Cuenta con un dispositivo móvil con conexión a Internet y lector de código QR. Mientras camina por la pequeña ciudad, desea conocer dónde se encuentran los puntos turísticos más relevantes, con el propósito de ir a visitarlos. Una vez que llega a cada lugar, accede a información detallada del mismo.

El prototipo para la medición abarca la última parte de la historia de usuario. A continuación se describen ambos prototipos desarrollados.

3.2.1. Prototipo: arquitectura híbrida

Para el desarrollo con esta arquitectura se utilizaron las siguientes herramientas desde el cliente: S.O. Android v3/4, IDE de Desarrollo Eclipse for Java Developers, Lenguaje de Programación Java., Android SDK (Kit de desarrollo de software), Http-Client (Para la conexión a servicios Web de tipo Restfull), ZXing (Procesamiento Multi-formato de imagen de código de barras 1D/2D para clientes Android).

Del lado del servidor, se usaron las siguientes tecnologías: S.O. Windows 2008 server, Servidor WEB IIS 7.0, IDE Visual Studio .NET, Lenguaje de Programación Visual Studio .NET WCF (Visual Basic .NET), Base de Datos de Puntos de Interés SQL SERVER EXPRESS 2008.

El usuario accede a la aplicación Android clickeando el icono de la misma en el menú principal del dispositivo. Al abrirse la aplicación se presenta la pantalla principal de la aplicación (ver figura 4).



Figura 4. Pantalla de aplicación Android Cliente.

El usuario tiene la posibilidad de escanear una etiqueta con un código QR o bien ingresar el Id (Identificador o Código) del POI. El usuario presiona el botón Scanner y la aplicación ejecuta el Barcode Scanner mostrando el Lector del mismo. Una vez enfocada la etiqueta QR, el Barcode Scanner decodifica el código QR leído y envía el dato a la aplicación Android, la cual consulta la aplicación WCF, a través de la interfaz http-client, y de esta manera obtiene la información detallada del POI identificado con la etiqueta QR leída.

3.2.2. Prototipo: arquitectura web

Para el desarrollo con esta arquitectura se utilizaron las siguientes herramientas: S.O. Windows 2003 server, servidor WEB IIS 6.0, IDE de Desarrollo Visual Studio.NET, Lenguaje de Programación ASP.NET MVC 3.5 (Visual Basic .NET). Del lado del servidor, se usaron las mismas que en la arquitectura híbrida.

El usuario accede, ejecuta el Barcode Scanner presionando el icono de la misma en el menú principal del dispositivo. Al abrirse la aplicación se presenta la pantalla principal de la aplicación, la misma varía de acuerdo al dispositivo donde se esté ejecutando.



Figura 5. Pantalla de aplicación Web, muestra información del POI.

El usuario ejecuta la función Scanner y a continuación se visualizará el Lector del mismo. Una vez enfocada la etiqueta QR, el Barcode Scanner decodifica el código QR leído y al ser éste un URL, abre el browser del dispositivo y envía el requerimiento de apertura de URL al mismo. Un vez que el browser ingresa a la Aplicación Web el mostrará la página web devuelta por esta última con los datos del POI (ver figura 5). De esta manera se obtiene la información detallada del POI identificado con la etiqueta QR leída.

3.3. Definición de los ambientes y casos de prueba

Los casos de prueba se definen teniendo en cuenta los siguientes aspectos: arquitectura, puntos de interés y ambientes de prueba.

Los prototipos se ejecutan en dos ambientes de testeo, los cuales se diferencian de acuerdo a la conectividad que se utiliza en cada uno.

En ambos ambientes participan componentes elementales comunes: el dispositivo, el servidor y el Access Point. En base a esto se diseñaron dos ambientes de prueba que poseen diferentes tipos de conectividad: Wi-Fi y 3G/GPRS. Se consideran estos dos principalmente debido a que influyen en forma diferenciada en el consumo de batería y en el tiempo de respuesta.

3.3.1. En relación a la arquitectura

En relación a la arquitectura, se ejecutan los dos prototipos, Arquitectura Híbrida y Web, en forma alternada y en otros en forma simultánea. Esto tiene un efecto más importante en el primer ambiente, ya que se podrá comparar a las dos arquitecturas en un mismo momento y la performance de las mismas no dependerá del horario de ejecución y de los cuellos de botellas que se puedan producir en la red de telefonía móvil y la conexión ADSL al servidor.

3.3.2. En relación a los puntos de interés

Se define un punto de interés que corresponde a un lugar de Termas de Río Hondo que tiene relevancia turística: Isla del Sol o Tara Inti. En general, los puntos de interés tienen la siguiente estructura: Título, Ubicación, Descripción, Administrador, Galería de imágenes.

Respecto al punto de interés Tara Inti, se diseñan tres casos de prueba, los cuales varían en el tipo y cantidad de información:

- PI-Caso1: solamente texto.
- PI-Caso2: texto y una imagen de alta resolución.
- PI-Caso3: texto y 3 imágenes de alta resolución.

3.4. Evaluación de la calidad de los prototipos

3.4.1. Definición de requisitos no funcionales

El Propósito fundamental es medir la eficiencia de las arquitecturas en base a dos variables: tiempo y recursos. En este proyecto, no se medirá throughput directamente, ya que no es necesario definir el tráfico en función del tiempo, dado que la necesidad es comparar la eficiencia en virtud del tráfico generado por las arquitecturas para una misma función y no el comportamiento de las mismas en relación con el canal de comunicación.

Tabla 3. Árbol de Requerimientos.

1. Eficiencia
1.1 Comportamiento en el tiempo
1.1.1 <i>Tiempo de respuesta en la consulta a un POI</i>
1.1.2 <i>Bytes Recibidos por la Aplicación Móvil en la consulta a un POI</i>
1.1.3 <i>Bytes Enviados por la Aplicación Móvil en la consulta a un POI</i>
1.2 Utilización de los Recursos
1.2.1 <i>Consumo de energía del CPU</i>
1.2.2 <i>Consumo de energía de la Pantalla</i>
1.2.3 <i>Consumo de energía de la interface 3G</i>
1.2.4 <i>Consumo de energía de la interface Wi-Fi</i>

Se prevé que una arquitectura que use una mayor cantidad de datos en una transacción utilizará más recursos del dispositivo impactando negativamente en la eficiencia de la aplicación. Pero, sí es necesario determinar el grado de consumo de energía de cada arquitectura en una tarea común en las aplicaciones de m-turismo como lo es la consulta a un POI. En la tabla 3 se detalla el Árbol de Requerimientos. Para la característica eficiencia, se identifican dos subcaracterísticas y, para cada una de ellas, sus atributos que serán medidos.

Se realiza una ficha para cada atributo donde se especifica la métrica y el método de medición que se utilizará. Se ejemplifica, en la tabla 4, la ficha del atributo *Tiempo de respuesta en la consulta de un POI*.

Tabla 4. Ficha de especificación de atributos.

Ficha de Especificación de Atributos		Cód.: 1.1.1
Característica	Eficiencia	
Subcaracterística	Comportamiento en el tiempo	
Categoría	Aplicaciones móviles	
Entidad	Aplicación de m-Turismo	
Nombre	Tiempo de respuesta en la consulta a un POI	
Definición	Cantidad total del tiempo en la consulta a un POI	
Objetivo	Brindar el tiempo total insumido desde que se realiza la consulta del POI hasta que se recibe la información del mismo en la pantalla del dispositivo.	

Tabla 5. Ficha de especificación de métricas.

Métrica	Tipo	Directa
Nombre	Tiempo de respuesta en la consulta de un POI	
Objetivo	Determinar la cantidad de tiempo en segundos insumido en una consulta a un POI	
Autor	PNajar	
Versión	1.0	
Herramienta	Cronómetro digital	
Método de Medición		
Nombre	Cálculo de tiempo	
Especificación	El cronometraje dará inicio en el momento en el que se selecciona el código QR obtenido por la lectura a través de la cámara del dispositivo del POI del aplicativo Barcode Scanner. Y finalizará cuando éste se visualice en su totalidad en la pantalla del dispositivo.	
Escala Numérica		
Representación	Continuo	
Tipo de Valor	Real	
Escala	Absoluta	
Unidad		
Nombre	Segundos	
Acrónimo	"	

3.4.2. Diseño de las mediciones

En este apartado se definen las métricas que permitirán obtener los valores que cuantifican a los atributos de la entidad Aplicación de m-Turismo. El resultado de esta etapa son las fichas de Especificaciones de Métricas. En la tabla 5 se presenta la ficha correspondiente al atributo *Tiempo de respuesta en la consulta de un POI*.

3.4.3. Diseño de la evaluación

Para obtener el grado de satisfacción brindado por los requerimientos es necesario definir un indicador que permita interpretar cada atributo o concepto calculable del árbol de requerimientos definido anteriormente. Esta etapa tiene como

entrada la ficha de especificación de atributos y métricas definidas primeramente y como salida las Fichas de Especificación de indicadores. Se definieron los indicadores elementales de cada atributo como también los indicadores globales de cada subcaracterística de la eficiencia.

4 CONCLUSIONES

Actualmente se están llevando a cabo las mediciones de los atributos, utilizando las métricas. Luego, se utilizarán dichos valores para mapearlos en los indicadores elementales, reconociendo en cada arquitectura los resultados "malos/marginales/aceptables". A partir de allí, se calcularán los indicadores globales de cada subcaracterística. Así se podrá evaluar cuál de las arquitecturas es la más eficiente.

En el futuro se publicarán los resultados finales de la evaluación de estos prototipos, más una análisis de la selección de estos atributos para medir la calidad de producto.

5 REFERENCIAS

1. Becker P., P. Lew & L. Olsina. *Specifying Process Views for a Measurement, Evaluation, and Improvement Strategy*, In: Advances in Software Engineering Journal, Vol. 2012, 27 pg., 2012.
2. Herrera, S. I., P. J. Najar Ruiz, E. Ledesma & S. Rocabado. *Sistema de Información Móvil para Turismo Receptivo*. Revista Gestao e Conhecimento. ISSN 1808-6594, 2012.
3. ISO, International Standard Organization, ISO/IEC 25000, Software Product Quality Requirements and Evaluation, 2011.
4. Lew P., L. Olsina L., P. Becker P. & L. Zhang. *An integrated strategy to systematically understand and manage quality in use for web applications*. In: Requirements Engineering, 2011.
5. Olsina L., F. Pappa F. & H. Molina. *How to Measure and Evaluate Web Applications in a Consistent Way*, In: Web Engineering: Modelling and Implementing Web Applications, Springer London, 2008.
6. Olsina L., P. Lew, A. Dieser & B. Rivera. *Updating Quality Models for Evaluating New Generation Web Applications*, In: Journal of Web Engineering, Rinton Press, US, 2012.
7. Olsina L., P. Lew, A. Dieser & B. Rivera. *Using Web Quality Models and a Strategy for Purpose-Oriented Evaluations*. Journal of Web Engineering, Rinton Press, 10 (4), 2011.
8. Olsina, L. & G. Rossi. *Measuring Web application quality with WebQEM*. MultiMedia IEEE, vol.9, no.4, 2002.
9. Piattini M, F. García F. & I. Caballero. *Calidad de Sistemas Informáticos*, AlfaOmega, Mexico, 2007.