Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 12, No.6, Mes Junio, 2019
ISSN: 2306-2495 | RNPS: 2343
Pág. 55-71
http://publicaciones.uci.cu

Tipo de artículo: Artículo original
Temática: Soluciones informáticas
Recibido: 12/03/2019 | Aceptado: 18/04/2019 | Publicado: 20/06/2019

Aplicación móvil para el análisis de la información captada en SIGEv3.0

Mobile application for the analysis of the information captured in SIGEv3.0

Daynelis Brito Morales [1*], Johan Bravo Borrell [2], Lijandy Jiménez Armas[3]

[1]Centro de Tecnologías y Gestión de Datos, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas. Cuba. dbmorales@uci.cu
[2]Centro de Tecnologías y Gestión de Datos, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas. Cuba. jbravo@uci.cu
[3]Centro de Tecnologías y Gestión de Datos, Facultad de Ciencias y Tecnologías Computacionales, Universidad de las Ciencias Informáticas. Cuba. ljimenez@uci.cu

*Autor para correspondencia: dbmorales@uci.cu

Resumen

La Universidad de las Ciencias Informáticas (UCI), para gestionar los procesos estadísticos, posee la aplicación informática Sistema Integrado de Gestión Estadística (SIGEv3.0), desarrollado en el Centro de Tecnologías de Gestión de Datos (DATEC). Esta aplicación es una base para el apoyo en los procesos de análisis de información mediante los reportes estadísticos y así facilita el proceso de toma de decisiones a los clientes del producto. Actualmente sigue carece de funcionalidades para visualizar la información estadística captada, y depende de un ordenador conectado a una red cableada. El presente trabajo de diploma tiene como objetivo desarrollar una Aplicación móvil para el análisis de la información captada en SIGE v3.0. El estudio realizado sobre aplicaciones que

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 12, No.6, Mes Junio, 2019
ISSN: 2306-2495 | RNPS: 2343
Pág. 55-71
http://publicaciones.uci.cu

se asemejan al funcionamiento de la solución, definió la estructura y funcionalidades básica de la herramienta a realizar, seleccionando Modelo Vista Presentador (MVP) como arquitectura del sistema y como metodología de software AUP-UCI, así como las herramientas y tecnologías para el desarrollo de la solución. Se definió una estrategia de prueba plasmando los niveles de prueba unidad, aceptación y sistema, así como tipos de prueba funcionales, y los métodos seleccionados son: prueba caja negra con la técnica partición equivalente y prueba caja blanca con la técnica camino básico, todo con el objetivo de encontrar y corregir posibles errores.

Palabras clave: Aplicación móvil; análisis de información; reportes estadísticos.

Abstract

The University of Informatics Sciences (UCI), to manage statistical processes, has the computer application Integrated System of Statistical Management (SIGEv3.0), developed in the Center of Data Management Technologies (DATEC). This application is a basis for the support in the processes of information analysis through statistical reports and thus facilitates the process of decision making to the customers of the product. At the moment this application lacks of functionalities to visualize the statistical information stored, and depends on a computer connected to a wired network. The purpose of the present diploma work is to develop a mobile application for the analysis of the information captured in SIGE v3.0. The study carried out on applications that resemble the operation of the solution, defined the basic structure and functionalities of the tool to be performed, selecting Model View Presenter (MVP) as system architecture and as software methodology AUP-UCI, as well as the tools and technologies for the development of the solution. A test strategy was defined, setting the test levels, unit, acceptance and system, as well as functional test types, and the selected methods are: black box test with the equivalent partition technique and white box test with the basic path technique, all with the goal of finding and correcting possible errors.

Keywords: *mobile app; information analysis; statistical reports.*

Introducción

El creciente desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) en las últimas décadas ha impulsado en las empresas principalmente, el desarrollo de nuevos estilos de trabajo. Incorporando a muchos de sus

procesos productivos el uso de las tecnologías y con ello el internet, lo que les permite aumentar la organización y productividad. El análisis estadístico resulta fundamental para la recolección y agrupamiento de datos de diversos tipos, para construir con ellos informes estadísticos. Estos informes permiten conocer y analizar el comportamiento sobre diferentes temas, siempre desde un punto de vista cuantitativo. Francia y los Estados Unidos fueron los pioneros en repensar la estadística en función de las computadoras. Mejoraron, adaptaron y crearon nuevos instrumentos para estudiar grandes volúmenes de datos: nuevas técnicas y herramientas gráficas (Lacourly, 2000).

Un hecho trascendental e interesante que llevó a la estadística su justificación teórica y a la vez sus propios métodos fue el progreso del cálculo de probabilidades. Dicho proceso, junto con la estadística, permite estudiar problemas donde intervienen fenómenos aleatorios. En la actualidad, la estadística, junto con el cálculo de probabilidades, constituyen una rama independiente de la matemática con aplicaciones en casi todas las ciencias: física, astronomía, biología, genética, medicina, agricultura, sociología, informática y otras. En estas se hacen predicciones, encuestas, controles de calidad. Es claro que la lista no es exhaustiva, también se aplican los métodos de la estadística al estudio de fenómenos "no medibles" tales como la lingüística y la literatura (Diaz, 2007).

Actualmente la muestra de las estadísticas o datos en gráficas, representa toda la información de modo esquemático y preparadas para los análisis o cálculos posteriores. Los gráficos son muy útiles para la comparación de información. Así mismo, es la exploración rápida de los datos para imprimir de forma visual su compresión global. Puede ser frecuente utilizar representaciones visuales complementarias que resumen los datos de estudio. Con estas representaciones, adaptada en cada caso a la finalidad informativa que se persigue, se trasmiten los resultados de los análisis de forma rápida, directa y comprensible para un conjunto amplio de persona (Bravo, 2015).

Para el control y muestra de la información, es necesario el uso de aplicaciones que gestionen sistemas de información, utilizados por entidades y empresas. Para dichas instituciones es de gran importancia tener concentrada en una plataforma informática, base de datos u otra forma o dispositivo de almacenamiento toda la información estadística relevante para la entidad. Esto les permitirá una gestión única de los datos y las bases para el análisis, las tendencias y estilos que puede tomar el negocio, teniendo en cuanta la recopilación o gestión de la información a mostrar o graficar (Scott, 2016).

El Sistema Integrado de Gestión Estadística (SIGEv3.0), desarrollado en el Centro de Tecnologías de Gestión de Datos (DATEC) en la Universidad de las Ciencias Informáticas (UCI), es una aplicación web que tiene como objetivo, la automatización de los procesos de gestión estadística y su ventaja principal es la rápida confección de resúmenes estadísticos en forma tabular, a través de los cuales se dan a conocer, los principales resultados en las diferentes esferas económicas y sociales del país. SIGEv3.0 tiene integrado el Generador Dinámico de Reportes

(GDRv1.8), que le brinda un conjunto de herramientas para el diseño, generación y visualización de reportes, mediante los cuales se organiza y exhibe la información contenida en la base de datos. Estas soluciones informáticas fueron desarrolladas utilizando el marco de trabajo Symfony v1.1.7 y ExtJS v3.4.

Actualmente los reportes estadísticos de SIGEv3.0 son una base para el análisis de información, permitiendo así una mejor comprensión y entendimiento de los datos. En esencia estos reportes se muestran en forma de tabla, donde a pesar de existe un conjunto de datos que son útiles y que tributan una mejor toma de decisiones, la forma que se visualizan no permite hacer rápidas comparaciones que pudieran ser más rápidas y fáciles empleando gráficos o tablas estadísticas. Tributando directamente en la toma de decisiones operativa y estratégica por la dirección de cada cliente del producto, entre los que se encuentra La Fiscalía General de la República (FGR), Grupo Empresarial de la Industria Ligera (GEMPIL) y la Oficina Nacional de Estadística e Información (ONEI).

Muchos de estos clientes con frecuencia necesitan consultar información útil, en áreas donde no existe una red cableada o inalámbrica que les permita conectarse directamente a la base de datos de SIGEv3.0. Además, esta información pudiera estar disponible en algún lugar donde no tuviera acceso a una red accesible, en su casa, en reuniones o eventos por lo que ayudaría al proceso de toma de decisiones.

Debido a la necesidad de dar solución a la situación planteada se identifica como **problema de la investigación**, ¿cómo visualizar la información estadística captada en el Sistema Integrado de Gestión Estadística (SIGE v3.0), para contribuir al proceso de toma de decisiones?

Para dar cumplimiento al problema planteado se define como **objeto de estudio**, la visualización de información estadística, enmarcado en el **campo de acción**, aplicación móvil que visualice información estadística para el Sistema Integrado de Gestión Estadística (SIGEv3.0).

Para dar solución al problema de investigación planteado se define como **objetivo general**, desarrollar una aplicación móvil, que permita contribuir al proceso de toma de decisiones a partir de la información captada en el Sistema Integrado de Gestión Estadística (SIGEv3.0).

Materiales y métodos o Metodología computacional

El proceso de desarrollo de software se apoya en el uso de diferentes herramientas y tecnologías, las cuales, unidas a la metodología seleccionada, conforman el ambiente de desarrollo de un sistema.

Metodologías de desarrollo de software

Una metodología de desarrollo de software se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de desarrollo de un sistema de informático. Suele estar documentada y promovida por algún tipo de organización ya sea esta pública o privada. Tienen como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto, exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se clasifican en dos clases: las metodologías tradicionales o robustas y las ágiles o ligeras (Pressman, 2010).

Para el desarrollo de la solución se utilizará la metodología definida por el centro DroidLab para el desarrollo de proyectos. La misma consiste en una versión desarrollada por la UCI basada en la metodología de desarrollo de software AUP.

Metodología de desarrollo de software AUP versión UCI

La Universidad de las Ciencias Informáticas (UCI) desarrolló una versión de la metodología de desarrollo de software AUP (Proceso Ágil Unificado), con el fin de crear una metodología que se adapte al ciclo de vida definido por la actividad productiva de la universidad. Esta versión decide mantener para el ciclo de vida de los proyectos la fase de Inicio, pero modificando el objetivo de la misma y se unifican las restantes fases de la metodología de desarrollo de software AUP en una sola, nombrada Ejecución y agregándose también una nueva fase llamada Cierre (Yero Tarancón, 2015). A continuación, se muestra una tabla con las fases de la metodología AUP-UCI:

Tabla.1: Fases de la variación de AUP para la UCI.

Fases AUP	**Fases Variación AUP-UCI**	**Objetivos de las fases (Variación AUP-UCI)**
Inicio	Inicio	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Elaboración Construcción Transición	Ejecución	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
	Cierre	En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto.

La metodología de software AUP-UCI a partir de que el modelado de negocio propone tres variantes a utilizar en los proyectos, como son: CUN (Casos de uso del negocio), DPN (Descripción de proceso de negocio) o MC (Modelo conceptual) y existen tres formas de encapsular los requisitos los cuales son: CUS (Casos de uso del sistema), HU (Historias de usuario), DRP (Descripción de requisitos por proceso), surgen cuatro escenarios para modelar el sistema en los proyectos, los cuales son:

- ✓ **Escenario No 1:** Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- ✓ **Escenario No 2:** Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- ✓ **Escenario No 3:** Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- ✓ **Escenario No 4:** Proyectos que no modelen negocio solo pueden modelar el sistema con HU (Sánchez, 2015).

A partir del análisis e investigación realizada, se utilizó la variante #1 (CUN) y el escenario #1 de la metodología AUP-UCI, debido a que dicha metodología es apropiada para proyectos pequeños permitiendo disminuir las probabilidades de fracaso, por ser un producto de fácil uso utilizando cualquier herramienta. Con la adaptación de AUP que se propone para la actividad productiva de la UCI se logra estandarizar el proceso de desarrollo de software. Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos. Se redujo a 1 la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11. De igual manera, también es la metodología que más utilidad tiene en la UCI por ser una versión desarrollada en dicha institución.

Lenguaje de modelado

El modelado es el diseño de las aplicaciones de software antes de codificar. Es una parte esencial de los grandes proyectos de software, y útil para proyectos medianos e incluso pequeños. Un modelo juega el papel análogo en el desarrollo de software (Ruiz, 2006).

UML v2.1

El UML v2.0 (del inglés Unified Modeling Language, en español Lenguaje de Modelado Unificado) es un lenguaje de modelado visual estándar destinado a ser utilizado para el modelado de procesos de negocio; y para el análisis, diseño e implementación de sistemas basados en software. UML es un lenguaje común para los analistas, arquitectos y desarrolladores de software, utilizado para describir, especificar, diseñar nuevos procesos de negocio. UML proporciona valiosa ayuda a los profesionales visualizando, comunicando y aplicando sus diseños (Booch, 2009).

Se utiliza UML v2.1 ya que proporciona ventajas en la representación del ciclo de vida de un software. Además, posibilita la comunicación sencilla y rápida entre el programador y los clientes del software que se desarrolla; brindando la posibilidad de que estos últimos puedan expresar su conformidad con el producto o las nuevas mejoras que desean ver introducidas.

Herramienta CASE

Son un conjunto de programas y ayudas a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software, permitiendo la creación de todo tipo de diagramas (Jacobson, 2011).

Visual Paradigm for UML v15.1

Visual Paradigm for UML v15.1 Standard Edition (VP-UML SE) es una plataforma de modelado diseñada para apoyar a los arquitectos de sistemas, desarrolladores y diseñadores de software. Además, acelera el proceso de análisis y diseño de aplicaciones complejas facilitando la construcción y visualización de diferentes **tipos de** diagramas UML dentro de los que se incluyen diagramas de casos de uso, de clases, entidad relación y otros (Rumbaugh, 2009).

Se escoge la herramienta Visual Paradigm For UML v15.1 por soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas.

Lenguaje de programación

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas (ALEGSA, 2009).

JAVA v8.0

Java es un lenguaje de programación multiplataforma orientado a objetos. Además, cuenta con una máquina virtual que permite ejecutar cualquier código compilado en Java. Actualmente puede ser usado de forma gratuita, además de ser de código abierto (Holgate, 2012). Se eligió Java v8.0 como lenguaje de programación debido a que es el lenguaje utilizado para la programación en Android. También presenta una amplia variedad de documentación, lo cual facilita el trabajo de los desarrolladores.

Herramientas de desarrollo integrado

Un IDE (del inglés Integrated Development Environment, en español Entorno de Desarrollo Integrado), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios (InformaticaSONs, 2011).

Android Studio v2.2.2

Android Studio es un IDE, basado en IntelliJ IDEA1(del inglés Integrated Development Environment for Android, en español Entorno de Desarrollo Integrado para Android) de la compañía JetBrains. Es un entorno dedicado exclusivamente a la programación de aplicaciones para dispositivos Android. Utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma (academiaandroid, 2014). Se eligió Android Studio v2.2.2 como IDE de desarrollo debido a que es el IDE oficial de la plataforma Android, además es de código abierto y gratuito. Por lo cual es la herramienta idónea para el desarrollo de la aplicación y el consumo de servicios desde la APK.

Sistema Gestor de Base de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la Base de datos (BD), el usuario y la aplicación. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de

[1] IntelliJ IDEA es un IDE para el desarrollo de programas informáticos.

abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos (Editorial McGraw-Hill, 2010).

SQLite v3.0

Es una biblioteca que implementa un SGBD transaccionales SQLv3 autocontenido, sin servidor y sin configuración. Su código es de dominio público y de libre uso. Es multiplataforma y sus BD pueden ser fácilmente portadas sin ninguna configuración o administración (Rómmel, 2012).

Se seleccionó SQLite como SGBD debido a que es de código abierto y multiplataforma. Además, SQLite es una tecnología cómoda para los dispositivos móviles debido a su simplicidad, rapidez y usabilidad. Las características citadas anteriormente hacen a SQLite el SGBD idóneo para emplear en el desarrollo de la aplicación.

Servicios web

Un servicio web es un aplicativo que facilita la interoperabilidad entre varios sistemas independientemente del lenguaje de programación o plataforma en que fueron desarrollados. Este debe tener una interfaz basada en un formato estándar como XML[2] (del inglés extensible Markup Language, en español Lenguaje de Marcas Extensible) o JSON[3] (del inglés JavaScript Object Notation, en español Objeto de Notación JavaScript (Barry & Associates, 2015).

Los servicios web pueden ser de dos tipos, SOAP O REST.

SOAP (del inglés *Simple Object Access Protocol*, en español Protocolo de Simple Acceso a Objetos) es un protocolo de mensajería XML extensible que forma la base de los servicios web proporcionando un mecanismo simple que permite el envío de mensajes entre aplicaciones. Un mensaje SOAP es una transmisión de una vía desde un emisor SOAP a un receptor SOAP, lo cual propicia que cualquier aplicación pueda participar en este intercambio como emisor o receptor, además de ser completamente independiente del protocolo de transporte subyacente (Mahan, 2011).

REST (del inglés *Representational State Transfer*, en español Trasferencia de Estado Representacional) es un estilo de arquitectura de software para sistemas hipermedias distribuido. REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen cómo los recursos son definidos y diseccionados (Taft, 2010).

[2] XML es un lenguaje de marcas utilizado para almacenar datos en forma legible.

[3] JSON es un formato de texto ligero usado para el intercambio de datos.

El sistema Integrado de Gestión Estadística (SIGEv3.0) cuenta actualmente con varias capas de servicios intermedios. La primera es un proyecto backend (solo servidor, denominado SIGEv4) el cual se encuentra en etapa de desarrollo inicial y abarca de momento solo la seguridad del sistema. La segunda es interna, implementada en el Módulo Generado Dinámico de Reportes (GDR) para la consulta o muestra de información. Analizando estas dos estructuras API-REST, no es factible la utilización de las mismas por la Aplicación móvil para el análisis de la información captada en SIGEv3.0 debido a que la versión actual de SIGE es la 3.0 y las funcionalidades que se pudieran reutilizar están contempladas en la versión 4 orientado a otra arquitectura. Partiendo de este análisis la solución que se desarrollará utilizará una capa de servicios, la cual permita la interoperabilidad de SIGEv3.0 con la Aplicación móvil para el análisis de la información captada en SIGEv3.0.

Atendiendo a lo explicado anteriormente sobre la versión actual de SIGEv3.0 se eligió el estilo de arquitectura de software para sistemas hipermedias distribuidos REST, que se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen cómo los recursos son definidos y diseccionados.

Patrones arquitectónicos

Los patrones arquitectónicos proporcionan un marco de referencia necesario para guiar la construcción de un software, permitiendo a los programadores, analistas y todo el conjunto de desarrolladores del software compartir una misma línea de trabajo y cubrir todos los objetivos y restricciones de la aplicación. Es considerada el nivel más alto en el diseño de la arquitectura de un sistema puesto que establecen la estructura, funcionamiento e interacción entre las partes del software (Escalante, 2016).

Se utiliza como patrón arquitectónico el MVP (del inglés *Model View Presenter*, en español Modelo Vista Presentador), el cual es un **patrón derivado del conocido MVC** (del inglés *Model View Controller*, en español Modelo Vista Controlador). El patrón MVP permite **separar la capa de presentación de la lógica de la misma**, de tal forma que los aspectos relacionados con el funcionamiento de la interfaz quedan separados de cómo se representa en pantalla. En Android existe un problema derivado del hecho de que las actividades están íntimamente acopladas tanto con la interfaz como con el acceso a datos. El patrón MVP independiza la vista de la forma de conseguir esos datos. Divide la aplicación en al menos tres capas distintas, permitiendo probar cada una de ellas de forma independiente:

- ✓ **Modelo:** representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos.

- ✓ **Vista:** muestra el modelo en un formato adecuado para interactuar. Maneja la presentación visual de los datos representados por el modelo.

- ✓ **Presentador:** Incluye las clases del negocio, que hacen de puente entre la vista y el modelo, incluyendo clases con métodos que alberguen la lógica de la herramienta. Siempre los artefactos de la capa Vista invocarán a las clases albergadas en esta capa, nunca accederán al modelo directamente. Las clases presentadoras contenidas en esta capa de negocio son las que albergan el control y gestión sobre todo el contenido referente a su CU (Escalante, 2016).

Patrones de diseño empleados en la solución

Un patrón de diseño es una descripción de la comunicación entre objetos y clases, personalizada para resolver un problema de diseño general en un contexto particular. Identifica clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Se presentan como pares de problema-solución con nombre, sugiriendo aspectos relacionados con la asignación de responsabilidades (Leiva, 2016).

Los patrones de diseño se caracterizan por:

- ✓ Representar soluciones técnicas a problemas concretos.
- ✓ Propiciar la reutilización.
- ✓ Representar problemas frecuentes.

Patrones de diseño GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos (Larman, 2004). El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar un software de manera eficaz. En el desarrollo de la herramienta se aplicaron los siguientes patrones GRASP:

Experto: este patrón es el encargado de asignar una responsabilidad al experto en información, es decir, la clase que posee la información necesaria y suficiente para ejecutar la responsabilidad asignada. En la solución, este patrón se evidencia en la clase BD; esta clase es la experta en interacción y comunicación con la base de dato.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. El mismo, es el encargado de asignar a las clases la responsabilidad de instanciar otra clase. Este patrón se pone de manifiesto en la clase Add_Tematica, que es la encargada de crear objetos de BD.

Alta Cohesión: cada elemento del diseño debe realizar una labor única dentro del sistema, lo cual expresa que la información que almacena una clase, debe ser coherente y estar lo más relacionada con ella posible. Este patrón es el encargado de asignar responsabilidades, de manera que la información que se almacena en una clase, sea la necesaria y esté bien delimitada. Este se evidencia en la clase tb_tematica, ya que esta describe únicamente los atributos necesarios para las temáticas.

Bajo acoplamiento: este patrón consiste en mantener las clases lo menos relacionadas posible, de forma tal que, al producirse un cambio en alguna clase, se tenga el mínimo de repercusión en las otras. Se evidencia en la clase GestionarTematica.

Controlador: es utilizado por todas las clases implicadas en la capa de presentación de la lógica del negocio. Esta posee la responsabilidad de controlar el flujo de eventos mediante las actividades correspondientes. Este patrón se evidencia en la clase GestionarTematica.java, pues esta, se encarga de controlar los eventos de las actividades Add_tematica, Modificar_tematica, Eliminar_tematica y Listar_tematica.

Patrones de diseño GoF

Los patrones GoF (*Gang of Four*) se utilizan en situaciones frecuentes debido a que se basan en la experiencia acumulada al resolver problemas reiterativos. Ayudan a construir software basado en la reutilización, a construir clases reutilizables (Pérez, 2004). En el desarrollo de la herramienta se aplicaron siguientes patrones GoF:

Adapter (Estructural): Adapta una interfaz, facilitando la implementación para que pueda ser utilizada por una clase que, de otro modo, no podría utilizarla. Esto se evidencia en la clase Details_Tematica, interfaz creada para poder definir los parámetros de cada temática de manera independiente; de otra forma se haría un poco engorroso este trabajo.

Singleton: permite restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Este patrón se evidencia en la clase VolleyS, la cual es una clase estática que para la realización de sus funciones hace uso de los métodos existente o creados por la librería volley-master.

Resultados

Interfaces del sistema

A continuación, se muestran algunas de las interfaces del sistema:

La aplicación desarrollada consta con una **interfaz para la autenticación del usuario**, en ella el usuario introduce el usuario y la contraseña. Además, cuenta con un botón donde puede comprobar su conexión a la base dato de

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 12, No.6, Mes Junio, 2019
ISSN: 2306-2495 | RNPS: 2343
Pág. 55-71
http://publicaciones.uci.cu

SIGEv3.0 mediante una capa de servicio. En la siguiente figura se muestra la interfaz para la autenticación del usuario:

Fig.1: Interfaz de autenticación o inicio de la solución.

La **interfaz de clasificaciones y elección de temáticas e indicadores**, el usuario autenticado seleccione, Temática, indicador asociado a esa temática, así como un periodo de tiempo, además define una leyenda con los colores rojo, amarillo y verde. En el botón Mostrar se selecciona la forma en que se desee visualizar la información seleccionada en Tabla Estadística, Gráfico de Barras y Grafico de Pastel. En la siguiente figura se muestra la Interfaz de clasificaciones y elección de temáticas e indicadores.

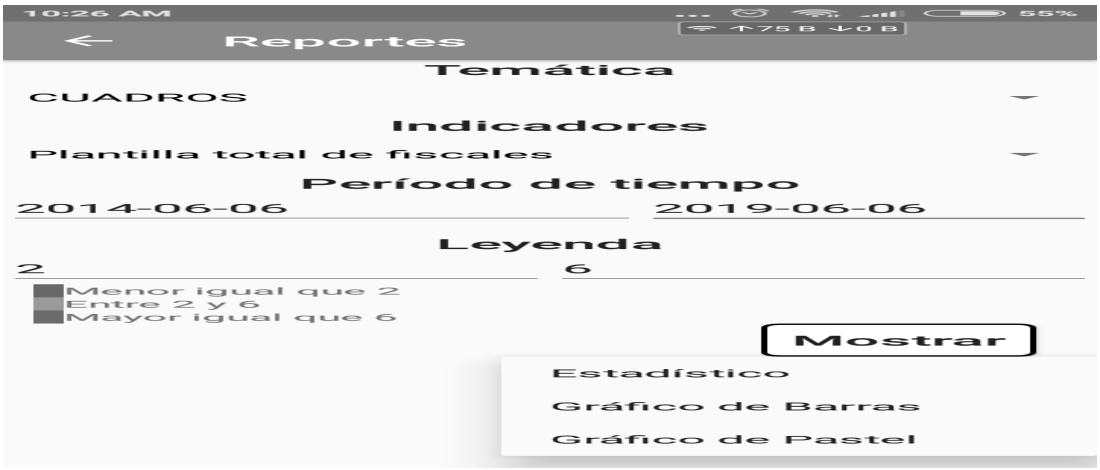

Fig.2: Interfaz de clasificaciones y elección de temáticas e indicadores.

Serie Científica de la Universidad de las Ciencias Informáticas
Vol. 12, No.6, Mes Junio, 2019
ISSN: 2306-2495 | RNPS: 2343
Pág. 55-71
http://publicaciones.uci.cu

La **interfaz de gráfica de barra**, muestra la información filtrada en la interfaz de clasificaciones y elección de temáticas e indicadores en una gráfica de barras, además de mostrar en la parte superior de la gráfica la información filtrada anteriormente (temática, indicador y periodo de tiempo). En la siguiente figura se muestra la Interfaz de gráfica de barra:

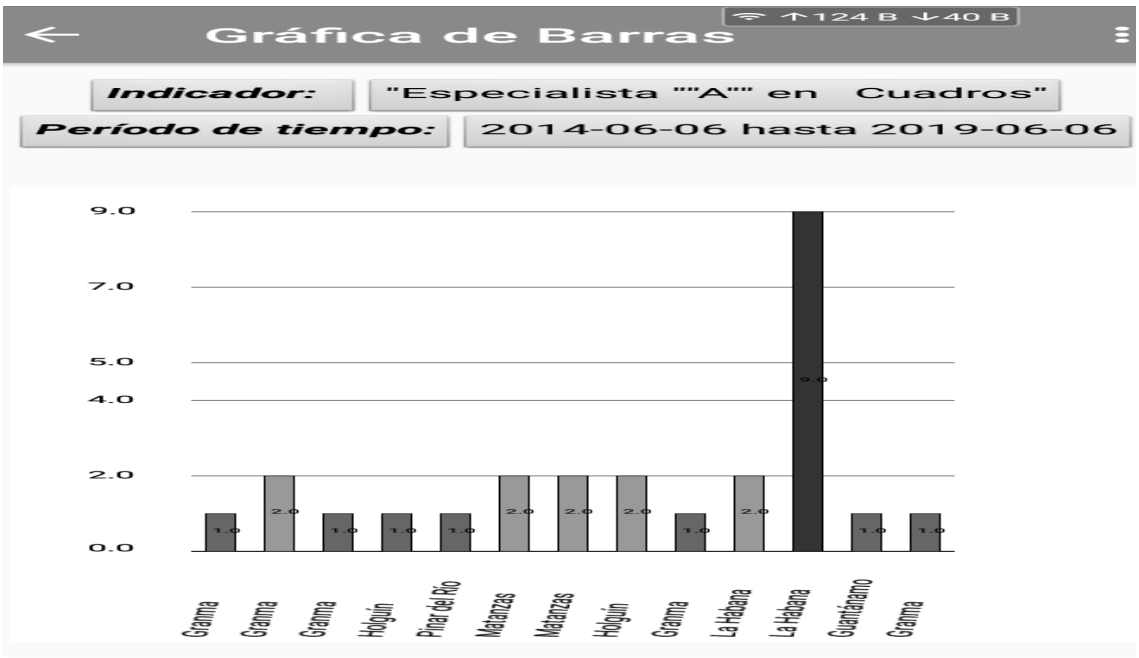

Fig.3: Interfaz de gráfica de barra

Conclusiones

Una vez concluida la investigación se arribó a las siguientes conclusiones:

- ✓ El estudio de los principales conceptos asociados a la investigación, permitió definir la estructura básica de la Aplicación móvil para el análisis de la información captada en SIGEv3.0, así como sus principales funcionalidades.
- ✓ El estudio del marco teórico de la investigación posibilitó la selección de la metodología, herramientas y tecnologías a utilizar para la implementación de la solución propuesta.

- ✓ La definición de los requisitos funcionales y no funcionales que debe cumplir el sistema permitió realizar un correcto análisis y diseño del mismo mediante el uso del patrón arquitectónico MVP, así como los patrones GRASP y GOF.
- ✓ La implementación de la solución desarrollada permitió dar cumplimiento a la totalidad de los requisitos de software identificados con el cliente durante la etapa de análisis.
- ✓ La solución fue validada a partir de la ejecución de pruebas unitarias, pruebas de sistema y pruebas de aceptación, las cuales permitieron comprobar el correcto funcionamiento de la Aplicación móvil para el análisis de la información captada en SIGEv3.0.

Referencias

academiaandroid. 2014. academiaandroid. *android-studio.* [En línea] 2014. [Citado el: 19 de 11 de 2018.] http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/.

ALEGSA. 2009. lenguajes-de-programacion. [En línea] 2009. [Citado el: 17 de 11 de 2018.] http://www.lenguajes-de-programacion.com/lenguajes-de-programacion.shtml.

Barry & Associates, Inc. 2015. www.service-architecture.com. *www.service-architecture.com.* [En línea] 2015. [Citado el: 5 de 11 de 2018.] http://www.service-architecture.com/articles/web-services/web_services_definition.html.

Booch, Grady . 2009. The Unified Modeling Language. [En línea] 2009. [Citado el: 17 de 11 de 2018.] http://www.uml-diagrams.org/.

Bravo, Darwin. 2015. https://es.slideshare.net. [En línea] Darwin Bravo, 5 de 3 de 2015. [Citado el: 1 de 6 de 2019.] https://es.slideshare.net/DARWINVD/la-representacion-grafica-de-estadistica.

Diaz, C., e I. de la Fuente. 2007. IEJME. [En línea] 2007. [Citado el: 2 de 4 de 2019.] https://www.iejme.com/volume-2/issue-3.

Editorial McGraw-Hill. 2010. www.cavsi.com. [En línea] 2010. [Citado el: 29 de 11 de 2018.] www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd.

Escalante, Lain Cárdenas. 2016. Tecnología & Desarrollo. *El patrón de arquitectura n-capas con orientación al dominio como solución en el diseño de aplicaciones empresariales.* s.l. : Tecnología & Desarrollo (Trujillo), vol. 11, no 1, p. 59-66., 2016.

GitHub. 2019. https://github.com. *https://github.com.* [En línea] 2019. [Citado el: 7 de 6 de 2019.] https://github.com/johnspice/jplot-android#m%C3%A9todos-publicos-que-ayudan-a-configurar-el-gr%C3%A1fico-de-pastel.

Historia, HiSoUR Arte Cultura. 2014. https://www.hisour.com. *HiSoUR Arte Cultura Historia.* [En línea] 11 de 3 de 2014. [Citado el: 31 de 5 de 2019.] https://www.hisour.com/es/chart-17607/.

Holgate, Colin. 2012. LiveCode Mobile Development Beginner´s Guide. 2012.

InformaticaSONs. 2011. http://www.sosinformatica.net. [En línea] 2011. [Citado el: 18 de 11 de 2018.] http://www.sosinformatica.net/evi/VisualBasic/guia_rapida/vb_guia_bd01.htm.

Jacobson, Ivar . 2011. umsl. [En línea] 2011. [Citado el: 17 de 11 de 2018.] http://www.umsl.edu/~sauterv/analysis/F08papers/View.html.

La Red Martínez, David Luis. 2007. http://exa.unne.edu.ar. [En línea] 2007. [Citado el: 1 de 6 de 2019.] http://exa.unne.edu.ar/informatica/SO/tfbasterretche.pdf.

Lacourly, Nancy. 2000. *Una pequeña historia de la estadística.* Chile : s.n., 2000.

Larman, Craig. 2004. *Uml y Patrones.* La Habana : Editorial Felix Varela, 2004.

Leiva, Antonio. 2016. DevExperto. [En línea] 28 de 2 de 2016. [Citado el: 1 de 4 de 2019.] https://devexperto.com/patrones-de-diseno-software/.

Mahan, Michael . 2011. http://searchsoa.techtarget.com. [En línea] 2011. [Citado el: 30 de 10 de 2018.] http://searchsoa.techtarget.com/tutorial/Simple-Object-Access-Protocol-SOAP-Tutorial&dt=-&s=-&r=DQE.

Pressman, Roger S. 2010. *Software Engineering. A practitioner´s Approach. 7th edition.* New York : McGraw Hill, 2010. ISBN 978-0-07-337597-7.

Rómmel, Filein . 2012. sqlite.org. [En línea] 2012. [Citado el: 29 de 11 de 2018.] http://sqlite.org/about.html.

Ruiz, Francisco - Lopez, Patricia. 2006. Lenguaje Unificado de Modelado - UML. [En línea] 2006. [Citado el: 2 de 4 de 2019.] https://ocw.unican.es/pluginfile.php/1403/course/section/1792/is1-t02-trans.pdf.

Rumbaugh, Jim . 2009. visual-paradigm. [En línea] 2009. [Citado el: 17 de 11 de 2018.] http://www.visual-paradigm.com/aboutus/.

Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI.* L a Habana, Cuba : s.n., 2015.

Scott, Kendall. 2016. http://datateca.unad.edu.co. [En línea] 21 de febrero de 2016. [Citado el: 1 de 11 de 2018.] http://datateca.unad.edu.co/contenidos/200609/exeuml/leccin_39_diagrama_de_clases_de_diseo.html.

Significados. 2019. https://www.significados.com. [En línea] 2019. [Citado el: 5 de 6 de 2019.] https://www.significados.com/estadistica/ .

Tecnologías-Información. 2018. Tecnologías-Información. *Tecnologías-Información.* [En línea] 2018. [Citado el: 5 de 6 de 2019.] https://www.tecnologias-informacion.com/visualizacion.html.

Vittone, José y Cuello, Javier. 2015. Capítulo 1: Las aplicaciones – Diseñando apps para móviles. [En línea] 2015. [Citado el: 25 de 11 de 2018.] http://www.appdesignbook.com/es/contenidos/las-aplicaciones/.

Yero Tarancón, Yixander . 2015. Xabal Excriba. *Gestor de Documentos Administrativos.* [En línea] 8 de abril de 2015. [Citado el: 18 de 11 de 2018.] http://excriba.prod.uci.cu/proxy/alfresco/api/node/content/workspace/SpacesStore/a622adab-eac5-4fb3-ba08-a266767fff5f/Metodologia%20UCI.pdf?a=true.