



LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Modelo de Control y Evaluación de Cuadros Maestros Determinados por el CNA, Gestionados en un Software

Control and Evaluation Model of Master Panels Determined by the CNA, Managed in a Software

Jhon Alexander Chapid Vela

jachapidv@libertadores.edu.co

Especialización Estadística Aplicada

Fundación Universitaria Los Libertadores

Ingeniero de Desarrollo. Universidad Mariana

Resumen

El Consejo Nacional de Acreditación (CNA), se ha implementado con el fin de que las instituciones y sus programas de pregrado y postgrado consoliden niveles educativos de alta calidad, con una mirada crítica en sus procesos, para generar estrategias de auto-evaluación constante. Uno de estos procesos son los Cuadros Maestros, los cuales deben diligenciarse periódicamente y que contiene la información histórica del programa. Para ello se desarrolló un software que permitirá gestionar los datos de manera más ágil y eficiente, centralizando la información y generando informes claros y concretos. Además de ello y ante la identificación de la necesidad de visualizar el comportamiento de los datos en diferentes periodos de tiempo de indicadores con la ayuda de R-studio y Scripts Markdown para la generación de archivos html y código R, que analizará y generar el comportamiento de los datos mediante gráficas, evidenciando como el programa ha evolucionado, con ello el interesado podrá tomar mejores decisiones, así mismo se tomó como metodología de desarrollo de software, la Metodología XP extreme programming, que posibilita el desarrollo cíclico con el fin de disminuir los tiempos de entrega y mejor adaptabilidad a los cambios continuos de requisitos funcionales. Los resultados obtenidos fueron el software y el análisis estadístico por medio de Scripts R-studio dentro del software con markdown, lo cual posibilita que herramientas opensource de estadística computacional puedan ser



integrados en cualquier sistema administrativo de información como una parte importante de la estadística dentro de las tecnologías y desarrollo de proyectos de software actuales.

Palabras clave: Marckdown, Metodologia Xp, R-studio, CNA,Software,Script, Html. Cuadros Maestros.

Abstract

The National Accreditation Council (CNA) has been implemented so that institutions and their undergraduate and graduate programs consolidate high quality educational levels, with a critical look at their processes, to generate constant self-evaluation strategies. One of these processes is the Master Charts, which must be filled out periodically and which contain the historical information of the program. For this, a software was developed that will allow to manage the data in a more agile and efficient way, centralizing the information and generating clear and concrete reports. In addition to this and before the identification of the need to visualize the behavior of the data in different periods of time of indicators with the help of R-studio and Markdown Scripts for the generation of html files and R code, which will analyze and generate the behavior of the data through graphs, showing how the program has evolved, with this the interested party will be able to make better decisions, likewise the XP Extreme Programing Methodology was taken as a software development methodology, which enables cyclical development in order to reduce the delivery times and better adaptability to continuous changes in functional requirements. The results obtained were the software and statistical analysis by means of R-studio Scripts within the software with markdown, which makes it possible for opensource computational statistics tools to be integrated into any administrative information system as an important part of statistics within the technologies and development of current software projects.

Keywords: Markdown, XP Methodology 2, R-studio, CNA, Software, Html, Master Paintings.

INTRODUCCIÓN

El Sistema Nacional de Acreditación es implementado en Colombia para fortalecer la calidad de la educación superior, y dar reconocimiento público a estándares de calidad que permitan una cultura de calidad continua, es así como el Concejo Nacional de Acreditación, en



adelante CNA, exige a las instituciones la gestión de cuadros maestros como herramienta que permita conocer y evaluar periódicamente un programa, los cuales deben contener la información periódica e histórica del mismo para la toma de decisiones relacionadas con su calidad; El diligenciamiento, alimentación y análisis de la información condensada en los cuadros maestros respecto a estudiantes, docentes, se realizan de forma manual en hojas de cálculo, al igual que el análisis estadístico y el cálculo de datos importantes para medir la calidad del programa.

En razón a lo anterior se formuló la siguiente pregunta ¿Cómo formular un modelo de control, análisis y evaluación oportuna de la información gestionada en los Cuadros Maestros y evaluación de indicadores soportado en un software? lo cual permitió a través de la gestión de información, el análisis de datos contenidos en cuadros maestros y la comparación y tendencias en diferentes periodos de tiempo de indicadores principales, la creación de un modelo que permite acoplar e intercalar los instrumentos computacionales, generando así las representaciones estadísticas y de probabilidad a través de un software que contiene la herramienta estadística R, y la generación de archivos *markdown* de forma dinámica, abstrayendo la información gestionada en el software de forma dinámica.

Metodología

El diseño y desarrollo de software se caracteriza actualmente por utilizar las técnicas y herramientas más completas para lograr un producto de calidad el cual permitirá tener consolidada la información y centralizada, de tal manera que pueda ser consultada en cualquier momento, como también observar y evaluar el comportamiento de la información *realtime* y realizar comparaciones periódicas de las variables e indicadores.

El estudio, desarrollo, e implementación propuesto, por tiempo y costos de desarrollo, está basado en metodologías ágiles, para ello se tomó como base la Metodología XP Y Cristal. La cual permite adaptarse a los equipos de desarrollo de pequeños grupos, permitiéndole a estos conocer el estado del proyecto en cualquier momento.

Así mismo, es ideal para proponer proyectos de desarrollo en periodos cortos de tiempo



con mucha flexibilidad para moldear los cambios frecuentes en tiempo de desarrollo.

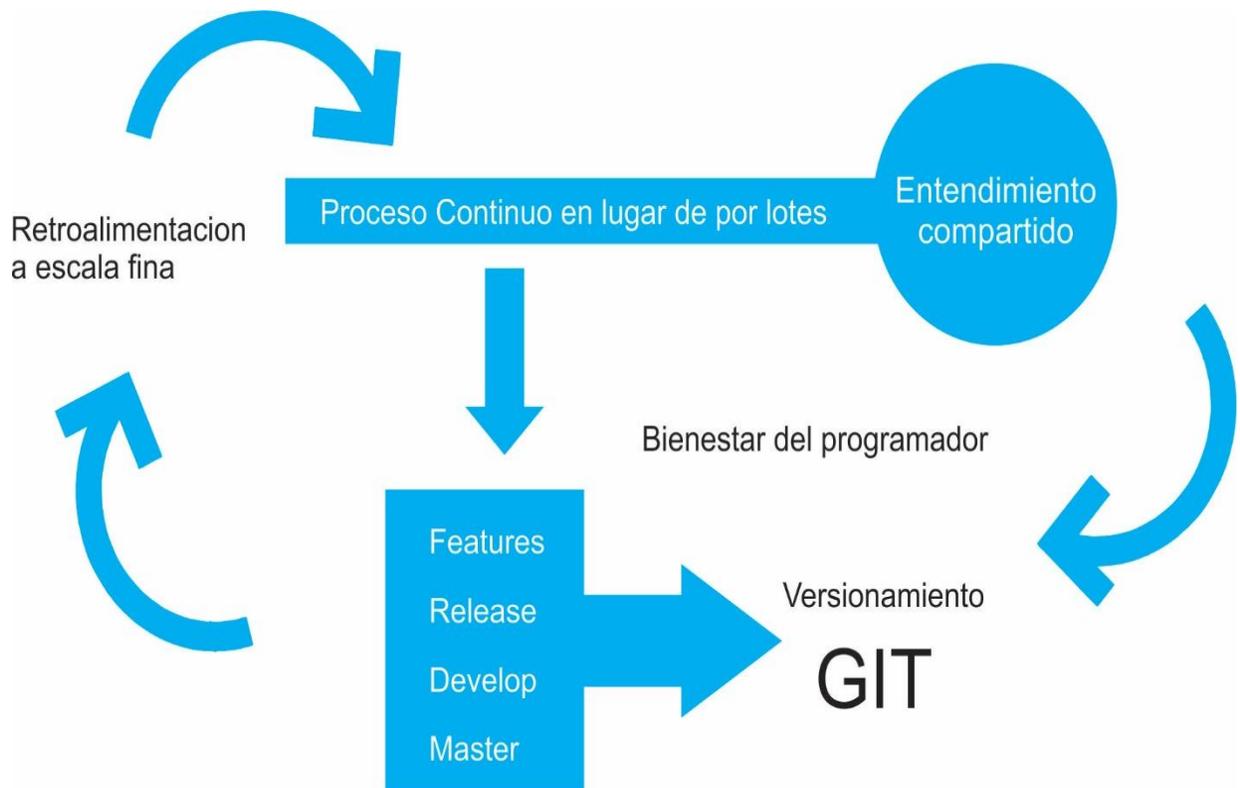


Figura 1. Modelo de desarrollo

La programación extrema, según López (2015) se tiene doce principios básicos, los cuales a su vez se agrupan en cuatro categorías a saber:

Retroalimentación a Escala Fina: esta fase incluye principios como los de realización de pruebas, proceso de planificación, el cliente en el sitio y programación en parejas.

Proceso Continuo en lugar de por lotes: este proceso permite la integración continua, refactorización (Evaluar el diseño del sistema a los largo de todo el proyecto y codificar si es necesario) y entregas pequeñas.

Entendimiento compartido: en esta tercera categoría se definen criterios como el de crear un diseño fácil, las tarjetas CRC (Clase, Responsabilidad y Colaboración) y la creación de la metáfora del sistema o historia completa.



Bienestar del programador: por último el bienestar del programador se rige por la filosofía que un programador cansado, exhausto crea código de mala calidad, por cuanto la autora menciona que se recomienda que los desarrolladores tengan 40 horas de trabajo a la semana y muy pocas horas extras de trabajo.

Respecto a la metodología Crystal, López (2015) afirma que se rige por principios que hacen su utilización eficiente, entre los cuales se pueden mencionar los siguientes:

Cada proyecto necesita un grado diferente de compensación (*Trade Off*), además se entiende que entre más pequeño sea el proyecto es mejor la forma de coordinación; por su parte cada uno de los proyectos necesita diferentes medios de comunicación. La autora antes mencionada dice que debe existir retroalimentación y comunicación efectiva, reduciendo así los problemas en entregas fallidas. Por último se debe considerar que los puntos dulces aceleran el desarrollo, se debe de contar con personas capaces y dedicadas a su trabajo, que se preocupen por hacer las entregas a tiempo y conforme a lo que el cliente solicita.

RESULTADOS

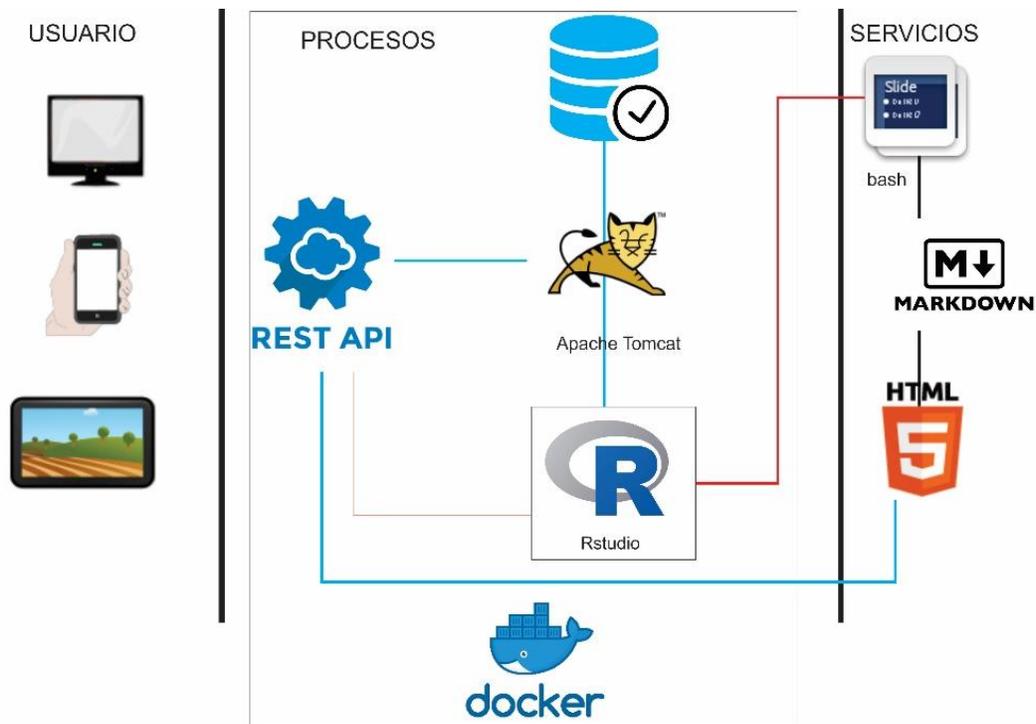
A continuación se describen los resultados obtenidos en cada una de las fases de desarrollo del proyecto:

Fase de retroalimentación a escala fina: En esta fase se planificó el diseño de cada módulo del software, los componentes o sub módulos que se integrarían y que permitiera la retroalimentación del sistema según la gestión de información realizada en los Cuadros Maestros. Se identificó las herramientas de desarrollo para el *frontend* (Desarrollo del lado del cliente), *Backend* (Capa de gestión de datos), Servidor (Versión de Linux), como también el proveedor de servicio en la nube y dominio para el posterior despliegue del proyecto.

Se diseñó la estructura de los Scripts *Markdown* y como serían accedidos y ejecutados para la posterior generación de archivos *html*, los cuales contienen el informe estadístico que es

presentado al usuario. Se identificó las librerías y paquetes de *R-studio* como: graficas, conexión a base de datos y funciones a utilizar para gestionar la información estadística. Así como su versión estable y compatibilidades. De igual realiza la selección de la información que retroalimentaría los módulos en el sistema y la base de datos con lo cual se optó por una base de datos relacional, como también se identificó que la configuración de contenedores *docker* agilizaría la virtualización de las herramientas y los procesamientos estadísticos.

Figura 2. Diseño de Implementación e integración de procesos



Se identificaron las fórmulas que permiten el cálculo de cada indicador e identificando las variables que contienen los cuadros maestros y que son necesarias para las proyecciones y tendencias de la información en los archivos *markdown*.

Proceso Continuo en lugar de por lotes: En esta fase se realizaron las pruebas de test por medio de lista de chequeo para evaluar el avance del proyecto y las correcciones que este tuviere, así mismo se optimizo el diseño en la implementación e integración del software, servidor de aplicaciones, *R.studio* y *Markdow*, como también la generación de prototipos continuos.



Se optimizó el diseño relacional de la base de datos, identificando las variables no necesarias y la actualización e inserción de la información contenida en los cuadros maestros hacia la base de datos, como también se realizaron las actualizaciones y configuraciones de las librerías necesarias y corrección de errores de dependencias en la gestión de scripts en *R-studio*, se identificó opciones de paquetes estadísticos de gráficos que permitirán una mejor experiencia para el usuario, según el informe del indicador.

Además de lo anterior las reuniones periódicas permitieron tener claro el actual y proseguir del proyecto, así mismo el control eficiente de las características del sistema realizadas en *localhost*, las cuales fueron probadas y su en código los comentarios respectivos para su posterior actualización en el servidor de sus contenedores respectivos.

Fase de entendimiento compartido: En esta fase se tuvo en cuenta el Diseño UX para brindar una mejor experiencia de usuario, la retroalimentación de las historias de usuario de acuerdo a cada módulo y sub módulo, evidenciando los requisitos, criterios de aceptación y descripción de proceso

Además se diseñó la integración de los componentes del software con la lógica computacional de *R-studio* para la generación estadística de los indicadores principales como (Índice de selectividad, absorción, deserción, relación docente Estudiante, Funciones Sustantivas, Tendencia en grupos de investigación, contratación docente y vinculación) a partir de la información gestionada y relacionada en los Cuadros Maestros.

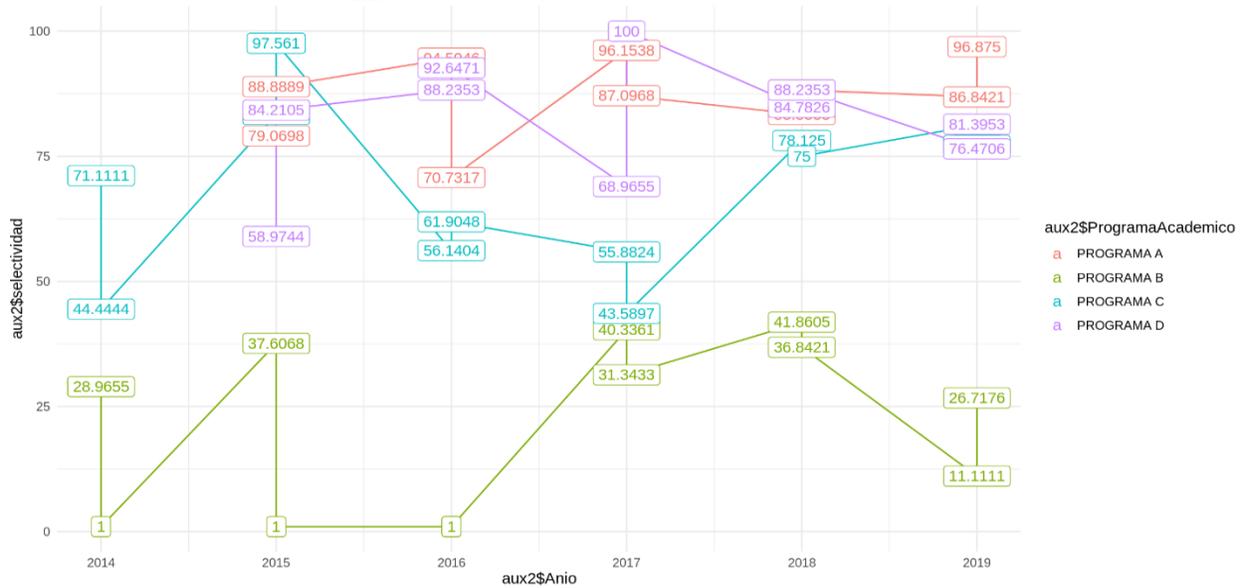


Figura 3. Ejemplo de generación de *R-studio*: Indicador Índice de Selectividad (Admitidos/Inscritos), programa A,B,C,D

Ejemplo: Análisis Información

“El índice de selectividad para el programa A, comparado con el Programa B, tiene un porcentaje bajo, por lo cual el programa A, tendrá que optimizar sus procesos de admisión”

Se puede observar un ejemplo de uno de los procesos de *Rstudio* por medio del software en el cual a partir de la información gestionada en cuadros maestros y mediante un Script *Markdown* se genera un informe estadístico que la persona interesada podrá hacer un análisis descriptivo del comportamiento de la información a partir de los periodos académicos en los últimos años.

Fase bienestar del programador: En esta fase se implementó GIT, que cuenta con unas ramas específicas para facilitar el almacenamiento ordenado y versionado de los archivos correspondientes a desarrollo, calidad y despliegue para controlar los estados del proyecto y su versionado como ayuda al control del código con ramas como:

- Master: Rama que contiene el código que saldrá a producción.
- *Develop*: Rama que contiene el código en construcción dentro que incluye las



funcionalidades terminadas.

- *Release*: Rama en la que se almacenan los componentes a ser probados.
- *Feature*: Rama temporal que almacena funcionalidades individuales y que desaparece cuando se concluye y se ramifica con la rama reléase.

Además se siguió el principio de planificación de codificación programada por faces o módulos con ayuda del versionamiento, para controlar el trabajo en equipo y la escalabilidad del proyecto sin agregar requisitos innecesarios que maximicen el tiempo de desarrollo, cumpliendo así, un tiempo dedicado al, diseño, desarrollo, pruebas y despliegue.

Conclusiones

Las nuevas tecnologías y lógica que se utiliza actualmente, posibilitan que cualquier sistema, lenguaje, software interactúe con procesos externos, lo cual incrementa la factibilidad de emplear estas tecnologías para procesos administrativos, de análisis y proyección en las empresas, y así ayuden a una mejor perspectiva al momento de tomar una decisión.

En el análisis de indicadores a partir de los scripts generados, permitieron tener una mayor dinámica al momento de visualizar la información y las tendencia que en estos se calculan, e indicar al programa a parte de la gestión de los Cuadros Maestros, como ha sido su evolución a partir de los últimos cinco años o por rango de fechas el actual de su programa, y así tomar decisiones más correctas para aumentar los niveles de calidad, promulgados por el CNA, para las futuras autoevaluaciones y planes de mejoramiento que este llevaré acabo.

Referencias Bibliográficas

Bustos Muñoz, J.A. (2014) Cuadros Maestros de Indicadores, División de Programación Académica. (En línea) <https://slideplayer.es/slide/4262758/>

Chapid, J y Luna, A (2015) *SAC - sistema orientado a la web para soportar el proceso de autoevaluación con fines de acreditación de los programas de pregrado de la Universidad de Nariño*. Trabajo de Grado, Universidad de Nariño San Juan de Pasto,



Colombia.

Chapid, J y Hubeymar, L (2016) *Sistema de Autoevaluación con fines de Acreditación Institucional de la Universidad de Nariño*. Universidad de Nariño, San Juan de Pasto, Colombia.

Chapid, J. (2020) Cuadros Maestros. (En línea) <http://cuadrosmaestros.tic-cloud.com.co/> (2020, septiembre 15)

Comité de Acreditación Facultad de Ingeniería – CAFI (2020). Cuadros Maestros. (En línea) https://rita.udistrital.edu.co/cafi/index.php?option=com_content&view=article&id=27&Itemid=315 (2020, septiembre 15)

Consejo Nacional de Acreditación (2020). *Guía de Procedimientos 01, Apreciación de Condiciones Iniciales para acreditación*. Bogotá Colombia (En línea). https://www.cna.gov.co/1741/articles-186376_Guia_Condiciones_Iniciales.pdf (2020, septiembre 15)

Consejo Nacional de Acreditación (s.f) *Lineamientos de acreditación* (En línea) <https://www.cna.gov.co/1741/article-186359.html> (2020, septiembre 15)

Docker Inc. (2020) Instalación de Tomcat (En línea) https://hub.docker.com/_/tomcat (2020, septiembre 15)

EcuRed (s.f), *Crystal*. (En línea) <https://www.ecured.cu/Crystal> (2020, septiembre 15)

Figuroa R.G, Solís CJ y Cabrera A (s.f) *Metodologías tradicionales vs. Metodologías ágiles*, Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación. Ecuador.

GitLab (s,f) *GitLab Development Kit* (en línea) <https://gitlab.com/gitlab-org/gitlab-development-kit#getting-started> (2020, septiembre 15)



Grolemund G y Wickham H. (2007) R para *Ciencia de Datos (Markdown)*, (En línea)
<https://es.r4ds.hadley.nz/formatos-de-r-markdown.html> (2020, septiembre 15)

Holmes, B., & T. Joyce, D. (2000). *Object-oriented programming with Java*. Sudbury. Jones and Bartlett Publishers.

Ingeniería de Software (2020) Metodologías Crystal en Métodos Agiles (En línea)
http://ingenieriadesoftware.mex.tl/59189_metodologia-crystal.html (2020, septiembre 15)

Largo, E. (2018) Aprende a configurar Apache Tomcat para desplegar una aplicación Java Web. (En línea) https://www.youtube.com/watch?v=_Ux6QOKKumk (2020, septiembre 15)

Letelier P y Penades C (2006) *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*, Revista Técnica Administrativa, Buenos Aires ISSN 1666-1680 (en línea) <http://www.cyta.com.ar/ta0502/v5n2a1.htm> (2020, septiembre 15)

López R.E (2015) Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales. Revista Tecnológica N° 8 (En línea)
<https://core.ac.uk/download/pdf/80296686.pdf> (2020, septiembre 15)

Meléndez Valladarez SM, Gaitan ME y Pérez Reyes NN. (2016) *Metodología Ágil de Desarrollo de Software Programación Extrema*, Universidad Nacional Autónoma de Nicaragua, Managua. (En línea) <https://repositorio.unan.edu.ni/1365/1/62161.pdf>, (20 enero de 2016)

Ministerio de Educación Nacional – SNIES (s.f) *Estadísticas históricas de la educación superior en Colombia*. (En línea) <https://snies.mineducacion.gov.co/portal/> (2020, septiembre 15)

Ministerio de Educación – SPADIES (s.f) *Tutorial SPADIES 3.0* (En línea)



<https://www.mineducacion.gov.co/sistemasinfo/spadies/Zona-de-Ayuda/268149:Tutorial-SPADIES-3-0> (2020, septiembre 15)

Ministerio de Educación Nacional (2017) *Educación Superior*. (En línea)
https://snies.mineducacion.gov.co/1778/articles-393225_boletin_dic_2017.pdf (2020, septiembre 15)

Navarro Cadavid A, Fernández Martínez JD y Morales Vélez J (2013) *Revisión de metodologías ágiles para el desarrollo de software*. Prospect. Vol. 11, No. 2 (En línea)
<https://dialnet.unirioja.es/descarga/articulo/4752083.pdf>, (2020, septiembre 15)

Oracle (2020) *Oracle Corporation and/or its affiliates*, Mysql. (En línea)
<https://www.mysql.com/> (2020, septiembre 15)

Platzi (s.f) Como crear un contenedor con Docker-Mysql y persistir la información (En línea)
<https://platzi.com/tutoriales/1432-docker/3268-como-crear-un-contenedor-con-docker-mysql-y-persistir-la-informacion/> (2020, septiembre 13)

Platzi (s.f) *Qué es Backend Frontend*. (En línea) <https://platzi.com/blog/que-es-frontend-y-backend/> (2020, septiembre 15)

Pressman, R. (2010). *Ingeniería del Software un Enfoque Práctico*. México, D.F, McGraw-Hill.

Rstudio – Server (2020) *RStudio Take control of your R code*. (En línea)
<https://rstudio.com/products/rstudio/#rstudio-server> (2020, septiembre 15)

Super-powered by Google (2020). *Code licensed under an MIT-style License*. Documentation licensed under CC BY 4.0. Versión 10.1.2-local+sha.31d0ee4cbf. (en línea)
<https://angular.io/>

Sistema Nacional de Acreditación (2013) *Lineamientos para la Acreditación de Programas de*



Pregrado, Bogotá Colombia. (En línea) https://www.cna.gov.co/1741/articles-186359_pregrado_2013.pdf (2020, septiembre 15)

SommerVille, I. (2005). *Ingenieras de Software*, Séptima edición. Madrid, Pearson Educación.

The Apache Software Foundation. (2019), *Apache Tomcat*. (En línea) <http://tomcat.apache.org/> (2020, septiembre 15)

Universidad de Alicante (2007) Configuración de Tomcat. Configuración de Aplicaciones Web (En línea) <http://www.jtech.ua.es/ayto/ayto2008/servd-web/sesion03-apuntes.html> (2020, septiembre 15)

Universidad de los Andes (2013) Cuadros Maestros. (En línea) <https://sites.google.com/site/acreditacionuniandes/home/programas/cuadros-maestros> (2020, septiembre 15)

Universidad de Nariño (2016) Documento Autoevaluación Institucional Sinopsis. (En línea): <https://www.udenar.edu.co/documentos/acreditacion/AutoevaluacionInstitucionalSinopsis.pdf> (2020, septiembre 15)

Universidad de San Buenaventura – Seccional Cali (2015) *Criterios Para El Cálculo De La Deserción Estudiantil En La Universidad De San Buenaventura Seccional Cali*. Cali Colombia. (En línea) https://www.usbcali.edu.co/sites/default/files/directiva_va-10.pdf (2020, septiembre 15)

Wells, D. (1999). *The Rules of Extreme Programming* (En línea) <http://www.extremeprogramming.org/rules.html> (2020, septiembre 15)

Wikipedia (2020) *Metodologías Agiles*. (En línea) https://es.wikipedia.org/wiki/Desarrollo_de_software (2020, septiembre 15)



Wikipedia (1999) Extrema o eXtreme Programming (En línea)
https://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema, (2020, septiembre 15)

Wikipedia (2020) *Diseño Relacional*. (En línea)
https://es.wikipedia.org/wiki/Modelo_relacional (2020, septiembre 15)