**Gestión de proyectos integradores**

Jonathan Arley Rodríguez Pacheco.

Carlos Daniel Salas Diaz.

Johamn Sebastián Rincón.

David Alejandro Reyes Caicedo.

Octubre 2019.

Institución universitaria Antonio José Camacho.

Ingeniería de sistemas.

Copyright © 2019 por Jonathan Rodríguez, Carlos Salas, Johamn Rincón & David Reyes. Todos los derechos reservados.

Contenido

[Introducción 4](#_Toc24627558)

[Problema de investigación 5](#_Toc24627559)

[Justificación 6](#_Toc24627560)

[Objetivos 7](#_Toc24627561)

[Stakeholders 8](#_Toc24627562)

[Solicitudes de usuario 9](#_Toc24627563)

[Estimación por puntos de caso de uso 12](#_Toc24627564)

[Arquitectura 16](#_Toc24627565)

[Metodología por implementar 17](#_Toc24627566)

[Patrones de diseño 19](#_Toc24627567)

[Herramientas de versionado de código 20](#_Toc24627568)

[Pruebas 25](#_Toc24627569)

[Tipos de mantenimiento 26](#_Toc24627570)

[Estándares en el modelo de datos y código fuente 27](#_Toc24627571)

[Proceso y Estándares de Documentación 28](#_Toc24627572)

[Diagrama de despliegue 29](#_Toc24627573)

[Diagrama de componentes 30](#_Toc24627574)

[Conclusiones 31](#_Toc24627575)

[Referencias 32](#_Toc24627576)

# Introducción

La institución universitaria Antonio José Camacho realiza una actividad semestral conocida como proyecto integrador, la cual tiene como objetivo aumentar el aprendizaje de sus estudiantes y ayudarles a familiarizarse con el entorno profesional en lo que respecta al proceso de desarrollo y elaboración de un proyecto.

En el siguiente documento de proyecto de curso se dará a conocer uno de los temas que corresponde al 5 semestre de ingeniería de sistemas, presentaremos cada uno de los debidos planteamientos que se realizaron para que el producto final pudiese ser de la mas alta calidad posible, además de cumplir con cada una de las expectativas del cliente (coordinador de proyecto integrador).

# Problema de investigación

Actualmente la gestión de los proyectos integradores de los estudiantes no está siendo efectiva, esto ha provocado varios problemas, uno de ellos es la desinformación que hay con respecto a los procesos que se deben realizar para cada una de las entregas del proyecto, sin mencionar las fechas de entrega definidas por los estudiantes como "un caos". La baja unión que hay entre los grupos formados para el proyecto también es un problema que destacar, debido a que, la gran mayoría ha experimentado la desinformación de primera mano, lo que conlleva al escepticismo por la "efectividad" de los proyectos integradores.

Retomando lo anteriormente mencionado, y en caso de que cada uno de estos problemas persistan, muchos de los estudiantes podrían manifestar su inconformidad con esta práctica, llevando al posible fin de la misma.

# Justificación

En la institución universitaria Antonio José Camacho, los proyectos integradores que se realizan en determinados semestres con el fin de aumentar los conocimientos y experiencias de los estudiantes, están principalmente orientados a dar solución a diferentes problemas de algunos de los procesos que se llevan a cabo en la universidad, se pudo identificar que estos problemas están generados por la falta de herramientas para el desarrollo en la universidad, entonces, con este proyecto se busca facilitar y mejorar la experiencia de cada uno de los actores que realizan algún proceso sobre los proyectos integradores, de tal manera que, pueda notarse un aumento de efectividad al ejercerse esta actividad, además de la satisfacción de los usuarios para con la solución.

# Objetivos

Este proyecto se llevará a cabo con la finalidad de resolver cada uno de los problemas anteriormente mencionados y de esta manera agilizar la gestión de los proyectos integradores.

Objetivo General:

Implementar por medio de una aplicación web la gestión de proyectos integradores de la Institución Educativa Antonio José Camacho.

Objetivos específicos:

1. Definir todos los posibles actores que pueden estar involucrados en la gestión de proyectos integradores.
2. Identificar los procesos que pueden llegar a realizar cada uno de los actores.
3. Implementar los procesos que cada actor realiza ofreciendo una herramienta (software) para ello.
4. Probar la herramienta, demostrar que funciona y cumple con los requerimientos establecidos.

# Stakeholders

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Descripción** | Expectativas |
| Coordinador de P.I | Aquel que se encarga de gestionar todo el proceso de los proyectos integradores (P.I). | Al coordinado de los P.I se le podrá facilitar el control de esta actividad. |
| Estudiante | Todos los estudiantes de 3 a 5 semestre que tengan mínimo 2 cursos relacionados con su carrera universitaria. | Los estudiantes adquirirán más motivación e interés por realizar esta práctica, debido a que contaran con un sistema que mejorará la experiencia. |
| Programadores | Se encargan de la elaboración del código | Adquirir experiencia. |
| Docente Asesor | Se encarga principalmente de asesorar a los estudiantes durante la realización de los P.I. | Facilitar el asesoramiento de los estudiantes y el control de los documentos. |
| Jurado | Se encargan de evaluar el proyecto final. | Optimizar el registro de las notas de cada grupo de P.I. |
| UNIAJC | La institución universitaria Antonio José Camacho en general. | Aumentar las expectativas por parte del público interesado de alguna manera en la institución. |

# Solicitudes de usuario

**Listado de requerimientos:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **¿Quién lo solicita?** | **Descripción del requerimiento** | **Funcional** | **Arq. Sig.** |
| 1 | Coordinador de P.I | El usuario ingresara sus datos y si estos son correctos accederá al sistema. | X | X |
| 2 | Programadores | El sistema contará con 3 módulos, en los cuales estarán definidos administrador, docente, jurado y estudiante con sus respectivos privilegios. | X | X |
| 3 | Coordinador de P.I | El Sistema debe permitir al administrador gestionar los grupos de proyectos integradores, así mismo, realizar cambios tales como: Cantidad de integrantes, Docente asesor, Semestre, Programa,  Jornada | X | X |
| 4 | Coordinador de P.I | El sistema debe permitir a los estudiantes realizar la inscripción a proyecto integrador; esto mediante un formulario el cual permitirá a los estudiantes ingresar el id del grupo al que desean pertenecer, esta inscripción tiene un límite determinado por el administrador. El sistema también debe permitir realizar transacciones de “CRUD” para el administrador. | X | X |
| 5 | Coordinador de P.I | El Sistema permitirá al administrador asignar el docente asesor para los grupos definidos | X | X |
| 6 | Coordinador de P.I | El sistema contará con un módulo el cual estará presente tanto para los estudiantes como para el administrador, este módulo le permitirá al estudiante llenar un formulario con la información respectiva de la asesoría una vez concluida la asesoría, se le notificará al profesor que realizó la asesoría en espera de una confirmación para tenerla en cuenta. El módulo permitirá al grupo revisar el estado de las asesorías (pendiente, aceptada, rechazada). | X | X |
| 7 | Coordinador de P.I | El sistema contendrá un repositorio con los campos que correspondan a la entrega (primera o última) y le permitirá al estudiante ir llenándolo, y podrá guardar sus avances, así mismo, para la primera entrega el estudiante tiene la opción de diligenciarlo por la aplicación o adjuntar un archivo pdf, mientras que en la segunda entrega el estudiante solo podrá adjuntar un archivo pdf, esto debido a que es un documento con parámetro de monografía. Una vez realizado el envió de alguna entrega, si se hizo antes de la fecha oficial establecida por el administrador, el grupo tendrá la opción de realizar algún cambio en dichas entregas, esto en caso de haberse cometido un error. | X | X |
| 8 | Coordinador de P.I | El sistema debe permitir al asesor realizar CRUD de la nota de primera entrega de cada grupo que tenga asignado | X | X |
| 9 | Coordinador de P.I | El sistema debe permitir al asesor realizar CRUD de la nota de la entrega final de cada grupo que tenga asignado | X | X |
| 10 | Coordinador de P.I | El sistema solo permitirá al administrador y al jurado realizar cualquier operación (asignar, modificar, eliminar) con respecto a la nota de la sustentación | X | X |
| 11 | Coordinador de P.I | El sistema debe permitir solo al administrador realizar cambios de las notas registradas, también permite postergar las fechas de entrega en casos especiales. | X | X |
| 12 | Coordinador de P.I | El sistema debe permitir enviar correo a los grupos de las siguientes formas: Por semestre: 4 semestre, 5 semestre  Por programa: ing sistemas, tecnología sistemas  Por jornada: mañana, noche  Masivo: todos  Grupo: S541, S441 | X |  |
| 13 | Coordinador de P.I | El sistema debe permitir consultar cada grupo por filtros:  semestre  programa  jornada  docente asesor  código de grupo | X | X |
| 14 | Coordinador de P.I | El sistema permitirá al docente consultar no solo las notas de sustentación por grupo si no también las notas que los jurados han dado individualmente a cada estudiante integrante de ese grupo de proyecto, esto por medio de un archivo Excel. | X | X |
| 15 | Coordinador de P.I | La información modificada sobre el proyecto integrador de cada semestre deberá reflejarse de forma inmediata para todos los usuarios. | X | X |
| 16 | Programadores | Se deberá poseer un respaldo de la base de datos actualizado en caso de cualquier perdida de datos que pudiese ocurrir. |  |  |
| 17 | Programadores | El usuario debe comprender fácilmente el sistema, esto gracias a una buena interfaz. |  |  |
| 18 | Programadores | En caso de ocurrir un error, el sistema debe reportarlos con mensajes claros e informativos para el usuario. |  | X |
| 19 | Programadores | El sistema debe poseer un diseño “Responsive”, esto con la intención de garantizar la adecuada visualización en distintos dispositivos. |  |  |
| 20 | Programadores | La base de datos será relacional y estará implementada en mysql, esto para mayor facilidad de uso. |  | X |

# Estimación por puntos de caso de uso



|  |  |  |  |
| --- | --- | --- | --- |
| **Actor** | **Num. De actores** | **Peso** | **Resultado** |
| Administrador | 1 | 3 | 3 |
| Estudiante | 1 | 3 | 3 |
| Docente-Asesor | 1 | 3 | 3 |
| Jurado | 1 | 3 | 3 |
|  |  | UAW | 12 |



|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de uso** | **Tipo** | **Peso** | **Resultado** |
| Gestionar usuarios | simple | 5 | 5 |
| Gestionar proyectos | simple | 5 | 5 |
| Gestionar Staff jurado | simple | 5 | 5 |
| Cronograma sustentación | simple | 5 | 5 |
| Gestionar asesorías | simple | 5 | 5 |
| Gestionar grupos | simple | 5 | 5 |
| Gestionar porcentajes calificación | simple | 5 | 5 |
| Enviar correos | simple | 5 | 5 |
| Gestionar formularios | simple | 5 | 5 |
| Asignar docente asesor | simple | 5 | 5 |
| Modificar notas y fechas de entrega | medio | 10 | 10 |
| Gestionar entregas | simple | 5 | 5 |
| Gestionar calificaciones | simple | 5 | 5 |
| Gestionar sustentaciones | simple | 5 | 5 |
| Diligenciar asesoría | simple | 5 | 5 |
| Diligenciar entrega | simple | 5 | 5 |
|  |  | UUCW | 85 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor técnico** | **Descripción** | **Peso** | **Impacto percibido** | **Factor Calculado** |
| T1 | Sistema distribuido | 2 | 1 | 2 |
| T2 | Rendimiento o tiempo de respuesta | 1 | 4 | 4 |
| T3 | Eficiencia del usuario final | 1 | 5 | 5 |
| T4 | Procesamiento interno complejo | 1 | 4 | 4 |
| T5 | El código debe ser reutilizable | 1 | 4 | 4 |
| T6 | Facilidad de instalación | 0,5 | 0 | 0 |
| T7 | Facilidad de uso | 0,5 | 5 | 2,5 |
| T8 | Portabilidad | 2 | 0 | 0 |
| T9 | Facilidad de cambio | 1 | 4 | 4 |
| T10 | Concurrencia | 1 | 5 | 5 |
| T11 | Características especiales de seguridad | 1 | 4 | 4 |
| T12 | Provee acceso directo a terceras partes | 1 | 2 | 2 |
| T13 | Se requiere facilidades especiales de entrenamiento a usuario | 1 | 4 | 4 |
|  |  | Factor Total Técnico 40,5 | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Factor-Ambiental** | **Descripción** | **Peso** | **Impacto percibido** | **Factor calculado** |
| E1 | Familiaridad con el modelo de proyecto utilizado  Familiaridad con UML | 1.5 | 5 | 7,5 |
| E2 | Personal tiempo parcial | -1 | 0 | 0 |
| E3 | Capacidad del analista líder | 0,5 | 4 | 2 |
| E4 | Experiencia en la aplicación | 0,5 | 3 | 1,5 |
| E5 | Experiencia en orientación a objetos | 1 | 5 | 5 |
| E6 | Motivación | 1 | 2 | 2 |
| E7 | Dificultad del lenguaje de programación | -1 | 5 | -5 |
| E8 | Estabilidad de los requerimientos | 2 | 5 | 10 |
|  | Factor Ambiental total | 23 | | |

1,4+(-0,03\*Factor ambiental total)

**Cálculo de UCP**

**Valores obtenidos:**

UUCP=97

TCF=1,005

ECF=0,71

UCP=97\*1,005\*0,71

UCP=69,21\*20(PF)

UCP=1384,287

Total de horas estimadas para una sola persona, pero tomando en cuenta que la cantidad de miembros del grupo es 4, podemos asegurar que este tiempo se verá reducido.

# Arquitectura

La arquitectura por utilizar en la implementación de este proyecto de software es CLIENTE-SERVIDOR, debido a que buscamos que los accesos, recursos y la integridad de los datos sean controlados por el servidor, además de aumentar la capacidad de clientes y servidores por separado. El mantenimiento de este es también una de las ventajas por las cual este tipo de arquitectura fue seleccionada.

Retomando lo anteriormente mencionado, es necesario adicionar que en un principio no se espera que el proyecto final sea inmediatamente utilizado por una gran cantidad de clientes, motivo por el cual la congestión del trafico no representa una variable negativa a tomar en cuenta a priori.



# Metodología por implementar

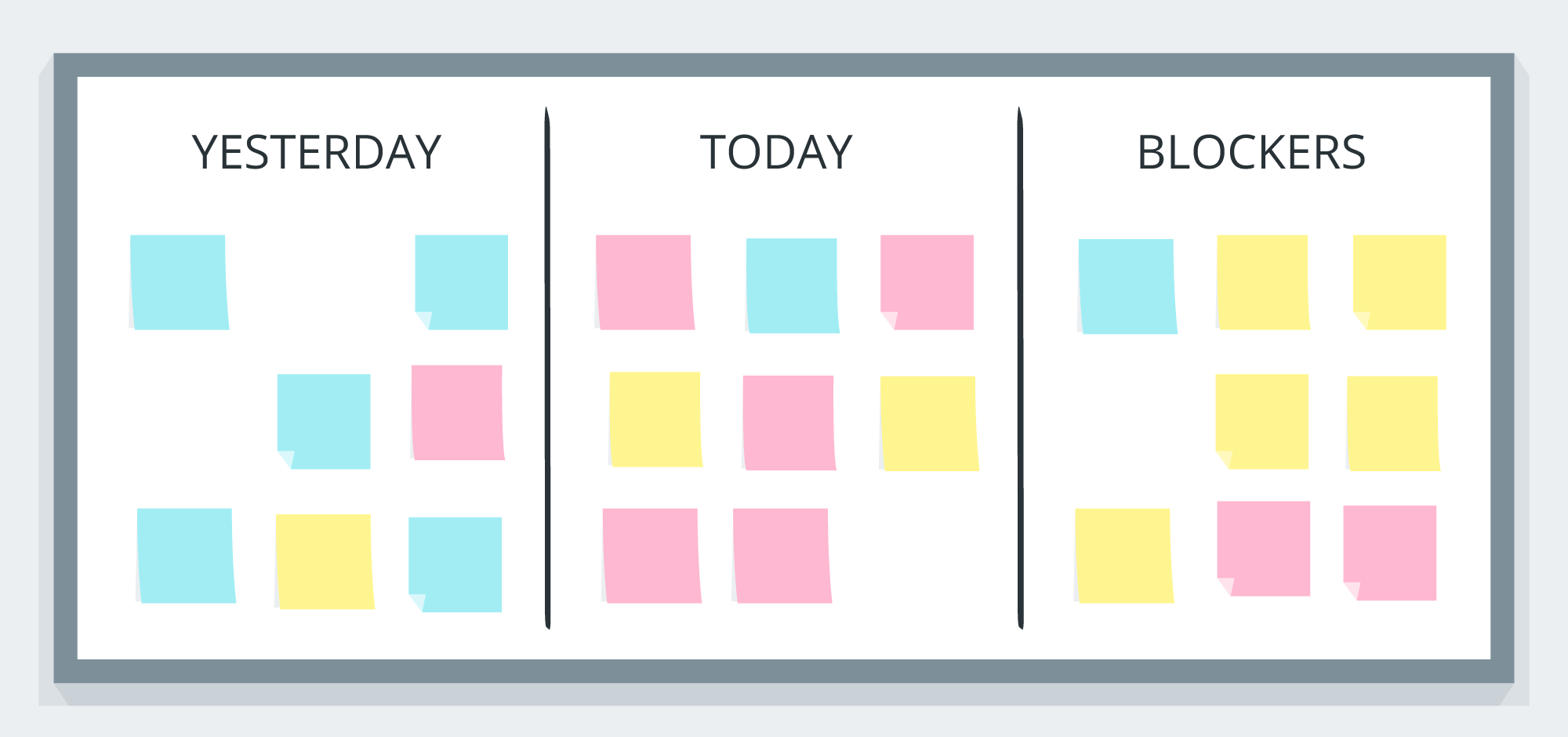
Se estableció que la metodología por implementar en este proyecto será Scrum, la cual es una de las metodologías agiles mas populares hasta hoy en día. Se basa principalmente en la premisa de realizar un proyecto en entregas parciales y regulares del producto. El desarrollo del producto se realiza de forma incremental y evolutiva, perfecto para entornos dinámicos y de constante cambio.

Los perfiles que componen la metodología Scrum son los siguientes:

* Product Owner: Es la persona encargada del éxito del producto desde el punto de vista de los clientes. Define los objetivos del proyecto y se focaliza en maximizar la rentabilidad del producto.
* Scrum Master: Es el coach del resto del equipo y quién lo ayuda a alcanzar su máximo nivel de productividad para garantizar el cumplimiento de objetivos.
* Scrum Team: está formado por todos los individuos necesarios para la construcción del producto en cuestión. Realiza un trabajo fundamental: persigue un objetivo común, con auto-gestión y una estructura auto-organizada.
* Stakeholders: son las personas a las que el proyecto les producirá algún beneficio (comerciales, dueños, directores).

**¿Cómo usaremos SCRUM?**

El equipo de proyecto se repartirá cada uno de los perfiles que componen la metodología SCRUM, se planea realizar cada una de las actividades modificando un poco la frecuencia de los daily scrum durante los sprints, estas reuniones se harán cada 3 dias y los sprints comprenderán un tiempo de 3 a 4 semanas como máximo y al igual que en la metodología original, se expondrá por parte de todo el equipo de proyecto el “¿Qué se hizo ayer?”, el “¿Qué se hará hoy? y el “¿Qué problemas se encontraron?”. Esta explicación del proceso de los daily scrum puede verse mejor representada en la siguiente imagen:

[[1]](#footnote-1)

Los artefactos como el producto backlog y el sprint backlog se definirán en base a pequeñas reuniones de asesoría con el cliente y los demás interesados, además es necesario mencionas que solo se realizara un sprint retrospective cuando el proyecto se encuentre en 50% de elaboración y otro al finalizar cuando el proyecto se encuentre al 100%.

# Patrones de diseño

Para la elaboración de este proyecto se estableció que el principal patrón de diseño a utilizar será Modelo Vista Controlador (MVC), el cual consiste en la separación de los datos y la lógica de negocio de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. El framework CodeIgniter ya posee este tipo de patrón de arquitectura de software establecido, por lo cual es considerado como el patrón principal a utilizar.

El patrón de diseño Fachada (Facade) tiene la característica de ocultar la complejidad de interactuar con un conjunto de subsistemas proporcionando una interfaz de alto nivel, la cual se encarga de realizar la comunicación con todos los subsistemas necesarios. Se considero utilizar este patrón para la implementación de este proyecto debido a la necesidad de interactuar con varios subsistemas para realizar operaciones concretas. En nuestro proyecto estaremos haciendo uso de GroceryCrud el cual funciona como si fuese el patrón de diseño Facade, facilitando el acceso por medio de una interfaz principal.

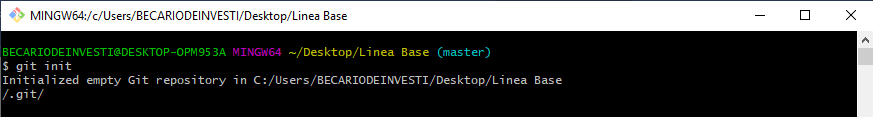
# Herramientas de versionado de código

Para llevar a cabo este proyecto con la mayor eficiencia posible se estableció que se haría uso de Git, el cual consiste en un software controlador de versiones muy útil para la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas poseen una gran variedad de archivos de código.

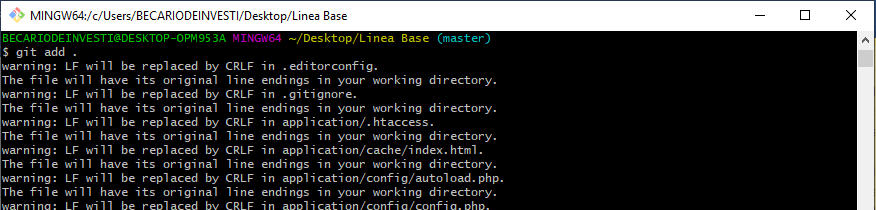
Retomando lo anteriormente mencionado, también se estableció el uso de GitHub, el cual consiste básicamente en una plataforma de desarrollo colectivo y colaborativo, es utilizado para el alojamiento de proyectos utilizando el sistema de control de versiones Git.

A continuación, se adjuntaron algunos de los procesos al utilizar estas herramientas de versionado

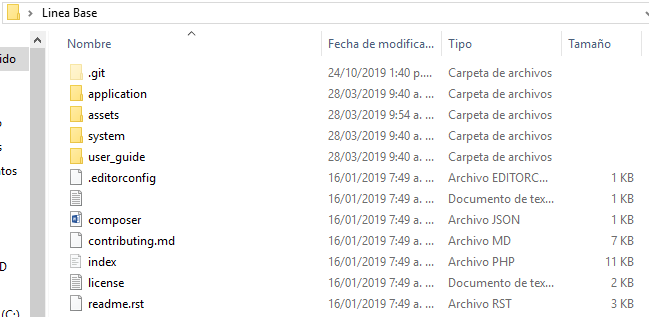
Inicializamos el proyecto con Git init



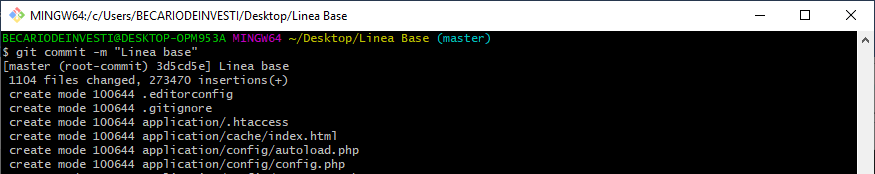
Agregamos la línea base Git add .



Los archivos de la línea base son CodeIgniter y GroceryCrud

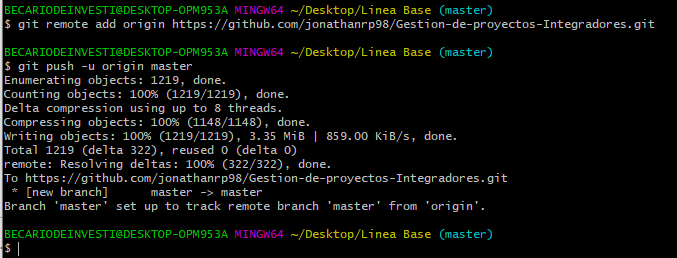


Realizamos el primer commit

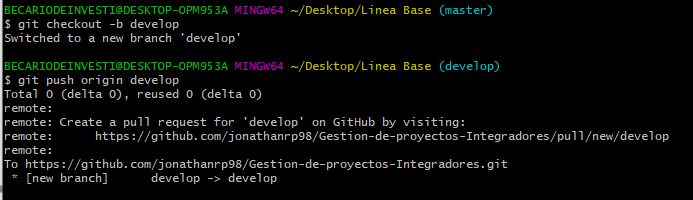


Anexamos la url donde se alojara el proyecto <https://github.com/jonathanrp98/Gestion-de-proyectos-Integradores.git>

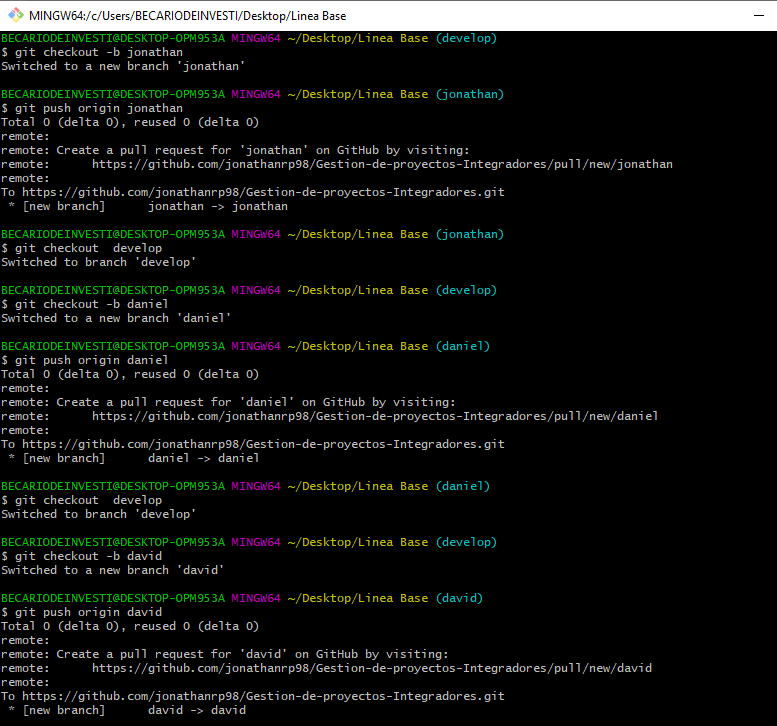
Y además hacemos el push a la rama principal

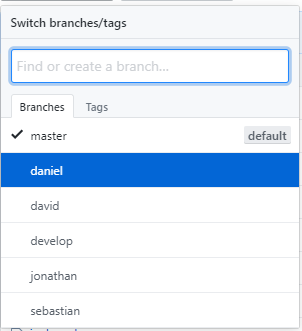


Creamos una rama develop y realizamos un git push origin develop para hacer una copia de los archivos que están en la rama master

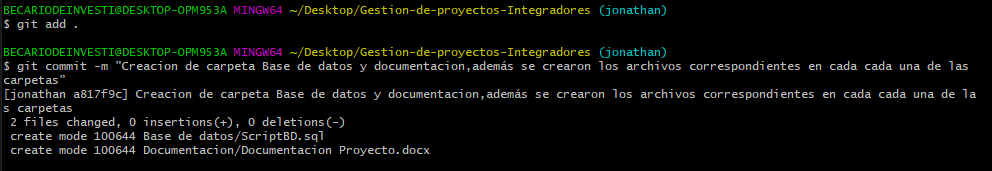


Creamos las ramas de los desarrolladores





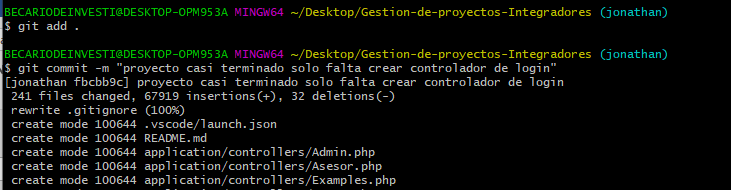
Creación de la carpeta base de datos y documentación, además se crearon los archivos correspondientes en cada una de las carpetas



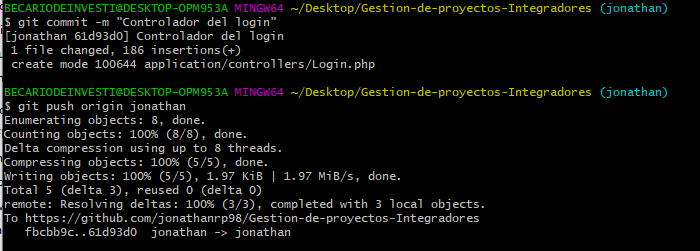
Agregamos los anteriores datos al repositorio



Agregamos los controladores, modelos y vista de las pantallas



Agregamos el controlador del login



# Pruebas

Las pruebas que se realizarán para este proyecto son las siguientes:

Para las pruebas de caja negra se utilizará la técnica de casos de prueba para cubrir todas las funciones que la aplicación es capaz de realizar, para esto deberemos tener en cuenta todo tipo de datos de entrada y salida, todos los elementos de diseño y cada clase de defecto. Este tipo de prueba se realizará a partir del escenario de casos de uso utilizado en este proyecto.

Para las pruebas de caja blanca se hará uso de PHPUnit el cual es un framework open source orientado a pruebas para cualquier código php, es necesario mencionar que para el uso de esta herramienta sera necesario que el código sea capaz de ser testeado completamente de manera unitaria, de lo contrario solo se podrán realizar pruebas de algunas de las funcionalidades de la aplicación. Tambien se utilizo el W3C Markup Validation Service, el cual permite a los usuarios analizar documentos HTML, verificar su estructura y si esta bien implementada validarlo.

Finalmente se realizarán pruebas de carga para evaluar cómo actúa nuestro sistema con una carga variable de usuarios, pero sin exceder los niveles esperados de la aplicación, consideramos esta prueba efectiva por ahora, debido a que no tomaremos en cuenta la posibilidad de que la cantidad de usuarios exceda la capacidad del sistema en su lanzamiento a producción.

# Tipos de mantenimiento

Los tipos de mantenimiento que se usaran para este proyecto son:

**Mantenimiento correctivo**

Se planea realizar todos los ajustes necesarios para corregir todos los errores que puedan llegar a surgir durante el testeo o uso de la aplicación, debido a que, al ser una aplicación con varias funciones, la posibilidad de que surja algún error es realmente alta.

Este mantenimiento se realizará haciendo uso de las pruebas de software y el testeo por parte de algunos colaboradores divididos en aquellos que tienen conocimientos de la ingeniería de sistemas y aquellos que no.

**Mantenimiento evolutivo**

La actividad de proyectos integradores (PI) puede llegar a sufrir varios cambios con el pasar de los años, de igual manera sucede con la tecnología, por lo tanto, consideramos importante realizar este mantenimiento cuando la aplicación pueda llegar a volverse obsoleta por el paso del tiempo. Este mantenimiento se realizará obteniendo información de las nuevas tecnologías y comparando cual es las mas indicada para implementar el cambio de nuestro proyecto.

**Mantenimiento perfectivo**

En caso de ser necesario se implementarán los cambios que el cliente no contempló al momento de la implementación, siempre y cuando estos cambios no provoquen un desvío excesivo de el objetivo inicial del proyecto.

# Estándares en el modelo de datos y código fuente

A nivel de base de datos tomamos parte del estándar que utiliza Oracle y realizamos cambios de mayor preferencia para el grupo.

* Nombre de la base de datos debe ser en mayúscula.
* Los nombres de las tablas deben ser en mayúsculas o CamelCase.
* Los nombres de las tablas deben ser en singular.
* Los nombres de las tablas deben ser descriptivos, sin importar que tan largos sean siempre y cuando sean soportados por la base de datos.
* Si la tabla tiene mas de 2 palabras se deben poner juntas separadas por un guion bajo, no usar espacios.

En cuanto al código, utilizamos una herramienta que genera documentación directamente del código fuente de PHP llamado phpDocumentor. Gracias a esta herramienta podemos proporcionar mas información integrada en el código fuente.

# Proceso y Estándares de Documentación

Nuestro estándar de documentación es estructurado por el siguiente proceso:

1. Creación de un borrador inicial.
2. Revisión del borrador inicial.
3. Incorporación de comentarios sobre errores en la documentación.
4. Corrección de texto.
5. Producción de borrador final.
6. Comprobar borrador final.
7. Arreglar texto.
8. Impresión del documento final.

Al implementar este proceso incluimos los siguientes estándares:

**ISO 26514**

Esta norma se refiere a los requerimientos de documentación de usuario para diseñadores y desarrolladores. Define los procesos de documentación desde el punto de vista de su desarrollador. Cubre la documentación como producto.

Esta norma apoya el interés de los usuarios de software en la documentación, que sea coherente, completa, precisa y utilizable. En nuestra documentación se planteo como objetivo crear una documentación que realmente fuese útil para cualquier persona que la leyera.

**IEEE 830**

Este estándar fue generado por un equipo de trabajo del IEEE, su finalidad es la integración de los requerimientos del sistema desde la perspectiva del usuario, cliente y desarrollador. En este proyecto se buscó respetar este estándar implementando los requerimientos desde varias perspectivas.

# Diagrama de despliegue

**Imagen que contiene texto

Descripción generada automáticamente**

# Diagrama de componentes

Imagen que contiene texto, mapa

Descripción generada automáticamente

# Conclusiones

La ingeniería de software nos ha enseñado a evitar cometer errores al momento de desarrollar un software, debido a que se estipula con antelación todo aquello que se espera realizar al implementar la aplicación.

El diseño de la arquitectura es parte fundamental de los principios de la ingeniería de software y es único gracias al hecho de que se organiza en función de los objetos y clases que se implementaran.

La implementación de la ingeniería de software es motor esencial para el éxito de cualquier proyecto a gran escala.

Los frameworks utilizados durante la elaboración del proyecto fueron de gran ayuda para mantener el debido orden de este, esto por medio del patrón de arquitectura de software (MVC).

# Referencias

* Proyectos Ágiles. (2019). *Qué es SCRUM*. [online] Available at: https://proyectosagiles.org/que-es-scrum/ [Accessed 3 Nov. 2019].
* Sistemas.com. (2019). *Cliente / Servidor*. [online] Available at: https://sistemas.com/cliente-servidor.php [Accessed 3 Nov. 2019].
* estructurales, D., clases, D., componentes, D., paquetes, D., despliegue, D., objetos, D., compuesta, D., perfiles, D., comportamiento, D., uso, D., secuencia, D., actividades, D., estados, D., comunicación, D., interacciones, D. and tiempos, D. (2019). *▷ Diagrama de componentes. Teoria y ejemplos*. [online] Diagramasuml.com. Available at: https://diagramasuml.com/componentes/ [Accessed 3 Nov. 2019].
* Ecured.cu. (2019). *Diagrama de despliegue - EcuRed*. [online] Available at: https://www.ecured.cu/Diagrama\_de\_despliegue [Accessed 3 Nov. 2019].
* Sites.google.com. (2019). *ISO/IEC 26514:2008 - SISTEMAS DE CALIDAD EN TI*. [online] Available at: https://sites.google.com/site/sistemasdecalidadenti/iso-iec-26514-2008 [Accessed 14 Nov. 2019].
* Fdi.ucm.es. (2019). [online] Available at: https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf [Accessed 14 Nov. 2019].

1. <https://www.range.co/blog/complete-guide-daily-standup-meeting-agenda> [↑](#footnote-ref-1)