

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS



PLAN DE DISERTACIÓN

TÍTULO:

**“DESARROLLO DE UN SISTEMA WEB PARA EL SOPORTE EN LAS
ACTIVIDADES DE GESTIÓN DE LA EMPRESA INOCIV CIA. LTDA.”**

JUAN CARLOS ESPINEL ORTIZ

JIMMY ALEXANDER NAVARRO VILLARRUEL

QUITO, 2018

DEDICATORIA

Dedicamos y agradecemos a nuestras familias por el constante apoyo brindado durante todo este proceso de aprendizaje en la universidad y en nuestro desarrollo como personas. Agradecemos de manera especial a nuestros profesores, lo cuales nos ayudaron y guiaron en el desarrollo de nosotros como profesionales.

Finalmente, un agradecimiento en especial a INOCIV CIA. LTDA por su apoyo durante el desarrollo de este trabajo. Además, gracias a nuestros amigos y compañeros que formaron parte de esta etapa que culmina el día de hoy.

ESQUEMA BÁSICO DE CONTENIDOS

1. Capítulo 1
 - 1.1 Antecedentes
 - 1.2 Justificación
 - 1.3 Objetivos
 - 1.3.1 Objetivos generales
 - 1.3.2 Objetivos específicos
2. Capítulo 2
 - 2.1 Descripción de la Empresa
 - 2.2 Misión
 - 2.3 Visión
 - 2.4 Objetivos INOCIV
 - 2.5 Situación Actual de la Empresa
 - 2.5.1 Negocio
3. Capítulo 3: Fundamentos teóricos
 - 3.1 Herramientas Para Desarrollar el Sistema
 - 3.1.1 PHP
 - 3.1.2 Base de Datos
 - 3.1.3 MySQL
 - 3.1.4 Aplicaciones Web
 - 3.1.5 Servidor De Aplicaciones
 - 3.1.6 HTML
 - 3.2 Metodología Para El Desarrollo Del Sistema
 - 3.2.1 Ingeniería de Software
 - 3.2.2 Metodología Ágil

3.2.3 Programación Extrema

3.2.4 Valores XP

3.2.5 El Proceso XP

3.2.6 El Debate XP

4. Capítulo 4: Metodología

4.1 Iteración 1

4.1.1 Planificación

4.1.2 Diseño

4.1.3 Codificación

4.1.4 Pruebas

4.2 Iteración 2

4.2.1 Planificación

4.2.2 Diseño

4.2.3 Codificación

4.2.4 Pruebas

4.3 Iteración 3

4.3.1 Planificación

4.3.2 Diseño

4.3.3 Codificación

4.3.4 Pruebas

4.4 Iteración 4

4.4.1 Planificación

4.4.2 Diseño

4.4.3 Codificación

4.4.4 Pruebas

4.5 Iteración 5

4.5.1 Planificación

4.5.2 Diseño

4.5.3 Codificación

4.5.4 Pruebas

4.6 Iteración 6

4.6.1 Planificación

4.6.2 Diseño

4.6.3 Codificación

4.6.4 Pruebas

4.7 Iteración 7

4.7.1 Planificación

4.7.2 Diseño

4.7.3 Codificación

4.7.4 Pruebas

5. Capítulo 5: Documentación

5.1 Manual de Usuario

5.2 Manual Técnico

6. Capítulo 6: Conclusiones y Recomendaciones

6.1 Conclusiones

6.2 Recomendaciones

7. Bibliografía

8. Anexos

8.1 Anexo 2:Manual de Usuario

8.1.1 Ingreso al Sistema

8.1.2 Proyectos

8.1.3 Estudios

8.1.4 Registrar Empleados

8.1.5 Registrar Usuarios

8.1.6 Sesiones Usuario

8.1.7 Cerrar Sesión

8.2 Anexo 3:Manual Técnico

8.2.1 Especificaciones Técnicas

RESUMEN

La empresa INOCIV fue creada en el 2015 con el fin de realizar diferentes estudios y obras a nivel de Ingeniería Civil, desde su funcionamiento ha tenido varios proyectos y estudios realizados a nivel nacional, actualmente no cuenta con una página web, ni con un sistema que agilice a Gestión de Proyectos y la Gestión de Empleados, todo lo lleva a cabo sin ningún orden, sin ningún récord histórico, sin ningún control de los empleados que se desempeñan en la empresa y además todo se encuentra en carpetas las cuales no agilitan cuando hay la necesidad de buscar un proyecto que ya se haya realizado cuando se presenta una auditoria o se necesite algún dato para un nuevo proyecto o estudio.

Ya que en la actualidad existe una rápida evolución de la tecnología y todo se ha llevado de lo físico que ya es obsoleto a lo digital, se siente en la necesidad de automatizar los procesos que son: Empleados y Proyectos, con un sistema el cual gestionara los proyectos, los estudios, a los empleados, los usuarios, las actividades que se encuentren en la realización de esta disertación.

INTRODUCCIÓN

Se va a desarrollar un software que va a ser aplicado por una empresa la cual no tiene procesos de control automatizados, siendo así una herramienta de gran utilidad. Cada vez que se desarrolla un software es necesario definir una metodología con la que se va a trabajar para cumplir de manera efectiva con el proyecto, esto ayudara a marcar las etapas

del desarrollo, comenzando desde los requerimientos de la empresa hasta las pruebas finales.

Aquí se abordarán conceptos computacionales para la elaboración del software, tales como: Ingeniería de Software, Metodología XP, entre otros.

1. CAPITULO 1

1.1 Antecedentes

Actualmente INOCIV es una empresa que se encarga de hacer auditoria en proyectos de Ingeniería Civil, dichos proyectos son registrados en papel y almacenados en carpetas, lo cual en un momento dado estaba bien, pero actualmente se necesita automatizar todo lo que se llevaba en carpetas además del registro de los empleados. INOCIV no cuenta con ningún proceso interno automatizado, lo que se lleva en computadora son hojas de Excel, en donde constan los contratos, costos.

1.2 Justificación

Actualmente INOCIV CIA. LTDA. no cuenta con una debida automatización de los procesos: Gestión de Empleados y Gestión de Proyectos, ya que todo se lleva a cabo en papel y carpeta, es decir, sin tener un acceso rápido con un control debido de los empleados y de los proyectos que ejerce dicha empresa.

Las ventajas de realizar el sistema para la empresa INOCIV CIA. LTDA. Permitirá una mayor eficiencia con una optimización de recursos, además mayor poder en la búsqueda y creación de proyectos que la empresa este ejecutando. Por este motivo se encontró la necesidad de implementar un sistema web que garantice el funcionamiento, el control, el acceso rápido a los datos que se requieran y con una interfaz amigable que cumpla con los requerimientos de la empresa.

Este tema de disertación es de utilidad para que la institución tenga un valor agregado, se realicen los procesos de control de una forma ordenada y eficiente para que posteriormente no se presenten problemas a futuro. Este tipo de sistema puede ser fácilmente adaptable para otro tipo de empresas que realicen las mismas actividades.

1.3 Objetivos: Generales y Específicos

1.3.1 Objetivos Generales

- Desarrollar un sistema web para el soporte en las actividades de gestión de la empresa: INOCIV CIA. LTDA.
- Cumplir con las normas y estándares para la realización del sistema web.

1.3.2 Objetivos Específicos

- Recopilar toda información necesaria sobre los proyectos y actividades que realiza la consultora INOCIV.
- Elaborar el documento de análisis de requerimientos de acuerdo a cierto estándar.
- Codificar y diseñar un sistema Web que facilita crear, organizar y compartir datos requeridos.
- Crear interfaces amigables que faciliten el manejo y utilidad de la aplicación web.
- Estructurar una base de datos que sea adecuada para un rápido almacenamiento.
- Diseñar la documentación respectiva del Sistema.

2. CAPITULO 2

2.1 Descripción de la empresa

INOCIV CJA. LTDA. Es una empresa que nace en el 2015 por la necesidad de emprender con nuevas estrategias para el cumplimiento de consultoría de obras civiles a nivel nacional, uniendo conocimiento multidisciplinarios de profesionales como arquitectos, ingenieros civiles, geógrafos, ambientales, eléctricos; garantizando la creación de nuevas oportunidades e ideas para el aprovechamiento de la tecnología a favor y beneficio del país en cuanto a los proyecto en los que la empresa participa

INOCIV CIA. LTDA. Cuenta con profesionales calificados para cada reto que se proponga en las diversas obras, prestando la atención necesaria a cada uno de sus clientes.

2.2 Misión

INOCIV CJA. LTDA. es una empresa consultora en el Ecuador de obras civiles, que garantiza a los gobiernos autónomos descentralizados del país y al público en general el cumplimiento de los más altos estándares en cuanto la construcción y la fiscalización en materia de ingeniería civil.

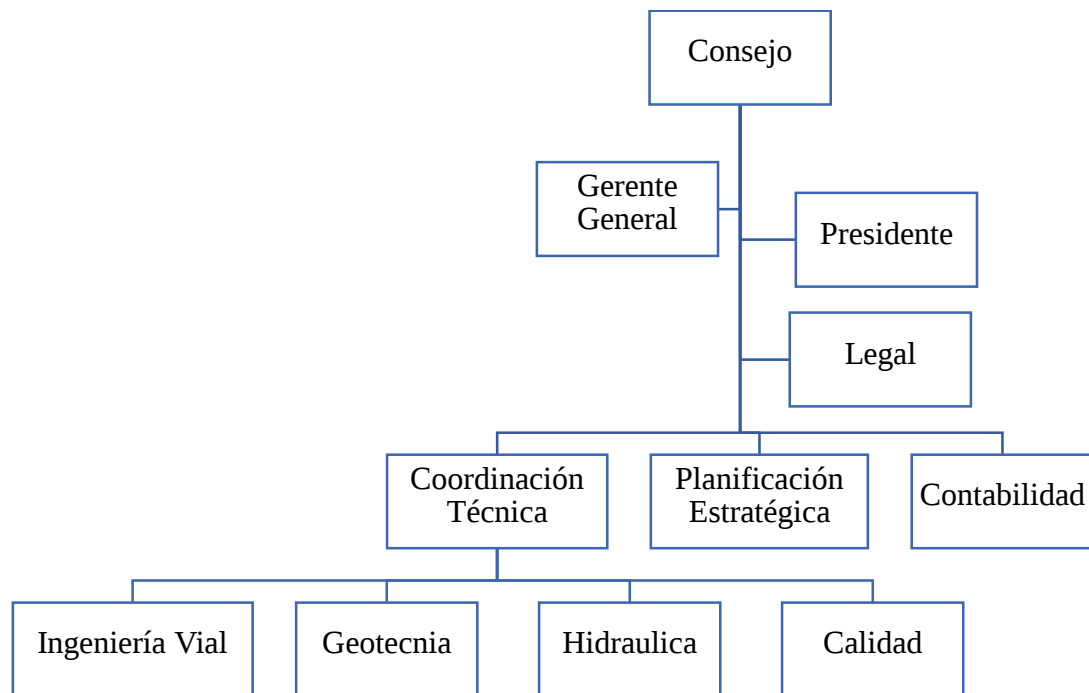
2.3 Visión

Ser una empresa líderes consultoría de obras civiles a nivel nacional garantizando a los beneficiarios de las mismas la seguridad de utilización y eficiencia de estas obras

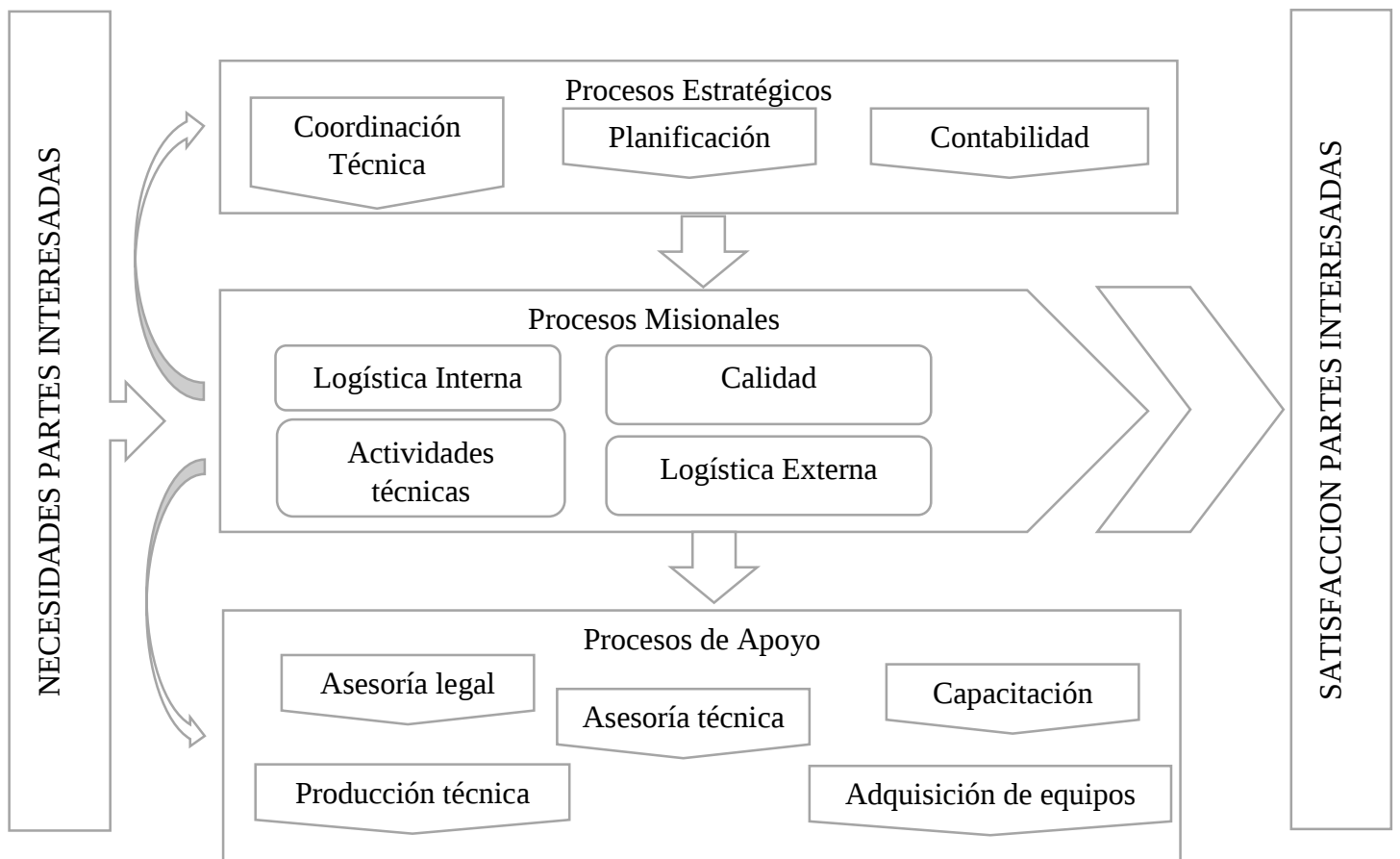
2.4 Objetivos INOCIV

- Utilizar estándares de construcción para garantizar la duración y eficacia de obras civiles
- Ofrecer nuevas alternativas de construcción en obras civiles.
- Crear nuevas oportunidades de aprovechamiento de recursos humanos económicos con alternativas amigables con el medio ambiente.
- Asumir las diversas necesidades que la ingeniería civil puede brindar a los usuarios a nivel nacional
- Definir estrategias de planificación para el cumplimiento de los diversos proyectos en plazos establecidos con la calidad necesaria.

ORGANIGRAMA INOCIV



MAPA DE PROCESOS



2.5 Situación actual de la Empresa

2.5.1 Negocio

La empresa aproximadamente en sus 3 años de creación ha logrado realizar y contribuir diversos proyectos a nivel nacional. Los gobiernos autónomos descentralizados que se han beneficiado con estos son los siguientes

- Gobiernos Autónomos descentralizados son Francisco de Orellana
- Gobiernos Autónomos descentralizados Provincia de Santo Domingo de los Tsáchilas

En cuanto a las empresas que han sido beneficiarias con proyecto a menor escala son:

- Consorcio Asociación E y S
- Esdicon Consultores CIA. LTDA.
- Empresa pública metropolitana de movilidad y obras públicas de Quito.

Por otro lado, se ha realizado informes y proyectos de menor escala para personas jurídicas en la construcción y estudios de suelos para diferentes obras.

El fuerte de la empresa es básicamente las obras civiles en cuanto a la consultoría de obras civiles en cuanto a la consultoría de obras viales diseño de pavimentos y puentes.

3. CAPITULO 3: Fundamentos Teóricos

Aquí se abordarán conceptos computacionales para la elaboración del software, tales como: PHP, Base de datos, MySQL, Aplicaciones web, Servidor de aplicaciones, HTML, Ingeniería de software, Metodología ágil.

3.1 Herramientas Para el Desarrollo del Sistema

3.1.1 PHP

PHP es un lenguaje de programación de alto nivel que se ejecuta y es interpretado en el servidor. Podemos encontrar este lenguaje en páginas importantes como Facebook, Wordpress, etc (Pavón y Llarena, 2016).

Un lenguaje servidor es aquel que se ejecuta en el servidor donde están alojadas las páginas, al contrario que otros lenguajes que son ejecutados en el propio navegador (Pavón y Llarena, 2016).

La principal ventaja es que, al ejecutarse el código en el servidor, todas nuestras páginas van a poder ser vistas en cualquier ordenador, independientemente del navegador que tenga. En cambio, el gran problema de que se ejecute el código en el navegador es que muchos navegadores no son capaces de entender todo el código, lo que presentaría errores al mostrar el resultado de las páginas (Pavón y Llarena, 2016).

PHP es un lenguaje incrustado en páginas HTML, es decir, es un lenguaje de programación que se introduce dentro de las páginas HTML. Su principal objetivo es hacer que desarrolladores de aplicaciones basadas en la web puedan escribir páginas que se generan dinámicamente de una forma sencilla y rápida (Millet y de la Iglesia, 2010).

3.1.2 Base De Datos

Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo

de organización como, por ejemplo, una universidad o un hospital (Marqués, 2011).

Una base de datos se puede percibir como un gran almacén de datos que se define y se crea una sola vez, y que se utiliza al mismo tiempo por distintos usuarios. Antes de existir las bases de datos, los programas debían manejar los datos que se encontraban almacenados en ficheros desconectados y con información redundante. En una base de datos todos los datos se integran con una mínima cantidad de duplicidad (Marqués, 2011).

De este modo, la base de datos no pertenece a un solo departamento, sino que se comparte por toda la organización. Además, la base de datos no sólo contiene los datos de la organización, también almacena una descripción de dichos datos. Esta descripción es lo que se denomina metadatos, se almacena en el diccionario de datos o catálogo y es lo que permite que exista lo que se denomina independencia de datos lógica-física, de la que se hablará más adelante (Marqués, 2011).

Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo: (Marqués, 2011).

- Agregar nuevos archivos vacíos a la base de datos (Marqués, 2011).
- Insertar datos dentro de los archivos existentes (Marqués, 2011).
- Recuperar datos de los archivos existentes (Marqués, 2011).
- Modificar datos en archivos existentes (Marqués, 2011).
- Eliminar datos de los archivos existentes (Marqués, 2011).
- Eliminar archivos existentes de la base de datos (Marqués, 2011).

3.1.3 MYSQL

MySQL es un sistema de gestión de bases de datos (SGBD) SQL, sus principales objetivos han sido la velocidad y la robustez. Es un SGBD sencillo y rápido que se adapta perfectamente a entornos en los que el volumen de datos sea del orden de megabytes (Millet y de la Iglesia, 2010).

- MySQL es un sistema de gestión de bases de datos

Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas cantidades de información en una red corporativa. Para añadir, acceder, y procesar los datos almacenados en una base de datos, necesita un sistema de gestión de base de datos como MySQL Server. Al ser los computadores muy buenos en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel central en computación, como aplicaciones autónomas o como parte de otras aplicaciones (Mysql, 2013).

- MySQL es un sistema de gestión de bases de datos relacionales

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones (Mysql, 2013).

3.1.4 Aplicaciones Web

Se denomina aplicación web al software que reside en un ordenador, denominado servidor web, que los usuarios pueden utilizar a través de Internet o de una

intranet, con un navegador web, para obtener los servicios que ofrezca (Pressman, 2010).

En los primeros días de la Red Mundial (entre 1990 y 1995), los sitios web consistían en poco más que un conjunto de archivos de hipertexto vinculados que presentaban la información con el empleo de texto y gráficas limitadas. Al pasar el tiempo, el aumento de HTML por medio de herramientas de desarrollo (XML, Java) permitió a los ingenieros de la web brindar capacidad de cómputo junto con contenido de información. En la actualidad, las webapps se han convertido en herramientas sofisticadas de cómputo que no sólo proporcionan funciones aisladas al usuario final, sino que también se han integrado con bases de datos corporativas y aplicaciones de negocios (Pressman, 2010).

Las webapps son una de varias categorías distintas de software. La gran mayoría de webapps presenta los siguientes atributos: (Pressman, 2010).

- Uso intensivo de redes: Una webapp reside en una red y debe atender las necesidades de una comunidad diversa de clientes. La red permite acceso y comunicación mundiales (por ejemplo, internet) o tiene acceso y comunicación limitados (por ejemplo, una intranet corporativa) (Pressman, 2010).
- Concurrencia: A la webapp puede acceder un gran número de usuarios a la vez. En muchos casos, los patrones de uso entre los usuarios finales varían mucho (Pressman, 2010).
- Carga impredecible: El número de usuarios de la webapp cambia en varios órdenes de magnitud de un día a otro. El lunes tal vez la utilicen cien personas, el jueves quizá 10 000 usen el sistema (Pressman, 2010).

- Rendimiento: Si un usuario de la webapp debe esperar demasiado (para entrar, para el procesamiento por parte del servidor, para el formado y despliegue del lado del cliente), él o ella quizá decidan irse a otra parte (Pressman, 2010).
- Disponibilidad: Aunque no es razonable esperar una disponibilidad de 100%, es frecuente que los usuarios de webapps populares demanden acceso las 24 horas de los 365 días del año. Los usuarios en Australia o Asia quizá demanden acceso en horas en las que las aplicaciones internas de software tradicionales en Norteamérica no estén en línea por razones de mantenimiento (Pressman, 2010).
- Orientadas a los datos: La función principal de muchas webapp es el uso de hipermedios para presentar al usuario final contenido en forma de texto, gráficas, audio y video. Además, las webapps se utilizan en forma común para acceder a información que existe en bases de datos que no son parte integral del ambiente basado en web (por ejemplo, comercio electrónico o aplicaciones financieras) (Pressman, 2010).
- Contenido sensible: La calidad y naturaleza estética del contenido constituye un rasgo importante de la calidad de una webapp (Pressman, 2010).
- Evolución continua: A diferencia del software de aplicación convencional que evoluciona a lo largo de una serie de etapas planeadas y separadas cronológicamente, las aplicaciones web evolucionan en forma continua. No es raro que ciertas webapp (específicamente su contenido) se actualicen minuto a minuto o que su contenido se calcule en cada solicitud (Pressman, 2010).
- Inmediatez: Aunque la inmediatez (necesidad apremiante de que el software llegue con rapidez al mercado) es una característica en muchos dominios de

aplicación, es frecuente que las webapps tengan plazos de algunos días o semanas para llegar al mercado (Pressman, 2010).

- Seguridad: Debido a que las webapps se encuentran disponibles con el acceso a una red, es difícil o imposible limitar la población de usuarios finales que pueden acceder a la aplicación. Con el fin de proteger el contenido sensible y brindar modos seguros de transmisión de los datos, deben implementarse medidas estrictas de seguridad a través de la infraestructura de apoyo de una webapp y dentro de la aplicación misma (Pressman, 2010).
- Estética: Parte innegable del atractivo de una webapp es su apariencia y percepción. Cuando se ha diseñado una aplicación para comercializar o vender productos o ideas, la estética tiene tanto que ver con el éxito como el diseño técnico (Pressman, 2010).

3.1.5 Servidor De Aplicaciones

Un servidor web es un programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada de HTTP). Un servidor web básico tiene un esquema de funcionamiento muy sencillo, ejecutando de forma infinita el bucle siguiente: (Mateu, 2012).

1. Espera las peticiones en el puerto TCP asignado (el estándar para HTTP es el 80) (Mateu, 2012).
2. Recibe una petición (Mateu, 2012).
3. Busca el recurso en la cadena de petición (Mateu, 2012).
4. Dirige el recurso a través de la misma conexión en la cual ha recibido la petición.
5. Vuelve al punto 2 (Mateu, 2012).

Un servidor web que siguiese el esquema anterior cumpliría los requisitos básicos de los servidores HTTP, aunque, eso sí, sólo podría servir ficheros estáticos (Mateu, 2012).

A partir del esquema anterior se han diseñado y construido todos los programas servidores de HTTP que existen, variando sólo el tipo de peticiones (páginas estáticas, CGI, Servlets, etc.) que pueden atender, en función de que sean o no multi-proceso, multi-hilados, etc. A continuación, detallaremos algunas de las características principales de los servidores web, que extienden, obviamente el esquema anterior (Mateu, 2012).

3.1.6 HTML

HTML es un estándar desarrollado con el objetivo de mostrar archivos de texto a un usuario agregando colores, estilos, diseños, esto hace que el archivo sea mucho más fácil en comparación con un archivo de texto plano (Luján, 2016).

El lenguaje HTML (hypertext mark-up language). Se trata de un lenguaje de marcas (se utiliza insertando marcas en el interior del texto) que nos permite representar de forma completa el contenido y también referenciar otros recursos (imágenes, etc.), enlaces a otros documentos (la característica más destacada del WWW), mostrar formularios para posteriormente procesarlos, etc (Mateu, 2012).

El HTML se creó en un principio con objetivos divulgativos. No se pensó que la web llegara a ser un área de ocio con carácter multimedia, de modo que, el HTML se creó sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro (Cornejo, 2012).

Resumen De Las Tecnologías

Herramientas	Tipo	Versión
MySQL (MariaDB)	Sistema de gestión de bases de datos	V 10.1.22
PHP	Lenguaje de programación	V 7.2.7
XP	Metodología ágil	
Apache	Servidor web	V 2.4.33
HTML	Lenguaje de marcas	V 5

3.2 Metodología Para El Desarrollo Del Sistema

3.2.1 Ingeniería De Software

La ingeniería del software es una disciplina de la ingeniería cuya meta es el desarrollo costeable de sistemas de software. Este es abstracto e intangible. No está restringido por materiales, o gobernado por leyes físicas o por procesos de manufactura. La noción de ingeniería del software fue propuesta inicialmente en 1968 en una conferencia para discutir lo que en ese entonces se llamó la “crisis del software”. Esta crisis del software fue el resultado de la introducción de las nuevas computadoras hardware basadas en circuitos integrados. Su poder hizo que las aplicaciones hasta ese entonces irrealizables fueran una propuesta factible. El software resultante fue de órdenes de magnitud más grande y más complejo que los sistemas de software previos (Sommerville, 2005).

La ingeniería de software incluye procesos, métodos y herramientas que permiten elaborar a tiempo y con calidad sistemas complejos basados en computadoras (Pressman, 2010).

A pesar de los grandes adelantos que ha habido en las técnicas de desarrollo de software durante los últimos treinta años, tanto la calidad del software como la productividad de su proceso de elaboración todavía no han alcanzado niveles plenamente comparables con los de otras tecnologías más antiguas. Todo esto, combinado con un aumento realmente espectacular de la demanda de software, ha provocado lo que se ha denominado la crisis del software (Campderrich, 2003).

En cuanto a la calidad, la causa principal de dificultades es la gran complejidad del software comparado con otros tipos de productos, que provoca, por ejemplo, que no sea posible, ni mucho menos, probar el funcionamiento de un software en todas las combinaciones de condiciones que se pueden dar. Y eso ocurre en una época en la que se da cada vez más importancia a la calidad en todos los ámbitos, al considerarla un factor de competitividad dentro de unos mercados cada vez más saturados y, por tanto, más exigentes. No es extraño, pues, que el tema de la calidad (y dentro de éste, cuestiones como la garantía de calidad y las certificaciones oficiales de calidad) adquiera una importancia creciente dentro de la ingeniería del software (Campderrich, 2003).

Por lo que respecta a la productividad, cabe decir para empezar que cualquier fabricación en serie tiene necesariamente una productividad mucho más elevada que la fabricación de un producto singular; pero, incluso, si la comparamos con otras ingenierías de producto singular, la productividad es claramente inferior. La complejidad del producto también puede ser una causa de este hecho, pero ciertamente no es la única. Un factor que tiene un peso realmente importante en la baja productividad es el hecho de que, a diferencia de las otras tecnologías, en un proyecto de software el desarrollo empieza tradicionalmente de cero (sólo

recientemente se utilizan fragmentos de software “prefabricados”) (Campderrich, 2003).

3.2.2 Metodología Ágil

Los métodos ágiles se han consolidado y se aplican en muchos campos, más allá de su nacimiento en el mundo del desarrollo del software. Algunas de ellas se centran en la forma más productiva de desarrollar aplicaciones informáticas mientras que otros buscan reflejar específicamente procesos de la empresa. La influencia de los métodos ágiles es tal que está afectando a la forma de contratar proyectos, a través de los contratos ágiles, que reflejan la flexibilidad y capacidad de adaptación aplicada a las relaciones formales entre empresas, de entre los métodos ágiles, destaca Scrum por su difusión y aceptación nacido antes del Manifiesto Ágil, se ha consolidado como el método más utilizado, y ha demostrado ser aplicable en toda clase de proyectos (Álvarez, de las Heredas y Lasa, 2012, p.24).

Esta forma ágil de desarrollar proyectos se fundamenta en cuatro puntos:

- Valorar a individuos y sus iteraciones (Álvarez, de las Heredas y Lasa, 2012).
- Valorar más el software que funciona (Álvarez, de las Heredas y Lasa, 2012).
- Valorar más la colaboración con el cliente (Álvarez, de las Heredas y Lasa, 2012).
- Valorar más la respuesta al cambio. (Álvarez, de las Heredas y Lasa, 2012)

3.2.3 Programación Extrema

Es un método ágil para el desarrollo de software muy útil a la hora de abordar proyectos con requisitos vagos o cambiantes. XP es especialmente útil si se aplica a equipos de desarrollo pequeños o medianos (Álvarez, de las Heredas y Lasa, 2012, p.21).

Es un método adaptativo, es decir, se ajusta muy bien a los cambios. Propone desarrollar el código de forma que su diseño, arquitectura y codificación permitan incorporar modificaciones y añadir una funcionalidad nueva sin demasiado impacto en la calidad del mismo. (Álvarez, de las Heredas y Lasa, 2012, p.21)

Por otro lado, XP es un método muy orientado hacia las personas, tanto a las que están creando el producto como a los clientes y usuarios finales (Álvarez, de las Heredas y Lasa, 2012, p.21).

Desarrollando como propone XP, se obtienen rápidamente resultados. Al trabajar con pequeñas iteraciones, se puede obtener con frecuencia comentarios del cliente, lo que tiene como resultado que el producto final cubra ampliamente sus expectativas y necesidades (Álvarez, de las Heredas y Lasa, 2012, p.21).

3.2.4 Valores XP

Beck define un conjunto de cinco valores que establecen el fundamento para todo trabajo realizado como parte de XP: comunicación, simplicidad, retroalimentación, valentía y respeto. Cada uno de estos valores se usa como un motor para actividades, acciones y tareas específicas de XP (Pressman, 2010).

A fin de lograr la comunicación eficaz entre los ingenieros de software y otros participantes (por ejemplo, para establecer las características y funciones requeridas para el software), XP pone el énfasis en la colaboración estrecha pero informal (verbal) entre los clientes y los desarrolladores, en el establecimiento de metáforas para comunicar conceptos importantes, en la retroalimentación continua y en evitar la documentación voluminosa como medio de comunicación (Pressman, 2010).

Para alcanzar la simplicidad, XP restringe a los desarrolladores para que diseñen sólo para las necesidades inmediatas, en lugar de considerar las del futuro. El objetivo es crear un diseño sencillo que se implemente con facilidad en forma de código. Si hay que mejorar el diseño, se rediseñará en un momento posterior (Pressman, 2010).

La retroalimentación se obtiene de tres fuentes: el software implementado, el cliente y otros miembros del equipo de software. Al diseñar e implementar una estrategia de pruebas eficaz, el software (por medio de los resultados de las pruebas) da retroalimentación al equipo ágil. XP usa la prueba unitaria como su táctica principal de pruebas. A medida que se desarrolla cada clase, el equipo implementa una prueba unitaria para ejecutar cada operación de acuerdo con su funcionalidad especificada. Cuando se entrega un incremento a un cliente, las historias del usuario o casos de uso que se implementan con el incremento se utilizan como base para las pruebas de aceptación. El grado en el que el software implementa la salida, función y comportamiento del caso de uso es una forma de retroalimentación. Por último, conforme se obtienen nuevos requerimientos como parte de la planeación iterativa, el equipo da al cliente una retroalimentación

rápida con miras al costo y al efecto en la programación de actividades (Pressman, 2010).

Beck afirma que la adhesión estricta a ciertas prácticas de XP requiere valentía. Un término más apropiado sería disciplina. Por ejemplo, es frecuente que haya mucha presión para diseñar requerimientos futuros. La mayor parte de equipos de software sucumben a ella y se justifican porque “diseñar para el mañana” ahorrará tiempo y esfuerzo en el largo plazo. Un equipo XP ágil debe tener la disciplina (valentía) para diseñar para hoy y reconocer que los requerimientos futuros tal vez cambien mucho, por lo que demandarán repeticiones sustanciales del diseño y del código implementado (Pressman, 2010).

Al apegarse a cada uno de estos valores, el equipo ágil inculca respeto entre sus miembros, entre otros participantes y los integrantes del equipo, e indirectamente para el software en sí mismo. Conforme logra la entrega exitosa de incrementos de software, el equipo desarrolla más respeto para el proceso XP (Pressman, 2010).

3.2.5 El proceso XP

La programación extrema engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas (Pressman, 2010).

Planeación: La actividad de planeación (también llamada juego de planeación) comienza escuchando (actividad para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software). Escuchar lleva a la creación de algunas “historias” (también llamadas historias del usuario) que describen la salida necesaria, características y

funcionalidad del software que se va a elaborar. Cada historia es escrita por el cliente y colocada en una tarjeta indizada. El cliente asigna un valor (es decir, una prioridad) a la historia con base en el valor general de la característica o función para el negocio. Después, los miembros del equipo XP evalúan cada historia y le asignan un costo, medido en semanas de desarrollo. Si se estima que la historia requiere más de tres semanas de desarrollo, se pide al cliente que la descomponga en historias más chicas y de nuevo se asigna un valor y costo (Pressman, 2010).

Los clientes y desarrolladores trabajan juntos para decidir cómo agrupar las historias en la siguiente entrega (el siguiente incremento de software) que desarrollará el equipo XP. Una vez que se llega a un compromiso sobre la entrega (acuerdo sobre las historias por incluir, la fecha de entrega y otros aspectos del proyecto), el equipo XP ordena las historias que serán desarrolladas en una de tres formas: 1) todas las historias se implementarán de inmediato (en pocas semanas), 2) las historias con más valor entrarán a la programación de actividades y se implementarán en primer lugar o 3) las historias más riesgosas formarán parte de la programación de actividades y se implementarán primero (Pressman, 2010).

Después de la primera entrega del proyecto (también llamada incremento de software), el equipo XP calcula la velocidad de éste. En pocas palabras, la velocidad del proyecto es el número de historias de los clientes implementadas durante la primera entrega. La velocidad del proyecto se usa para: 1) ayudar a estimar las fechas de entrega y programar las actividades para las entregas posteriores, y 2) determinar si se ha hecho un gran compromiso para todas las historias durante todo el desarrollo del proyecto. Si esto ocurre, se modifica el contenido de las entregas o se cambian las fechas de entrega final (Pressman, 2010).

A medida que avanza el trabajo, el cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas. Entonces, el equipo XP reconsidera todas las entregas faltantes y modifica sus planes en consecuencia (Pressman, 2010).

Diseño: El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después. XP estimula el uso de las tarjetas CRC como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Las tarjetas CRC (clase-responsabilidad-colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Las tarjetas CRC son el único producto del trabajo de diseño que se genera como parte del proceso XP (Pressman, 2010).

XP estimula el rediseño, técnica de construcción que también es un método para la optimización del diseño. Fowler describe el rediseño del modo siguiente:

Rediseño es el proceso mediante el cual se cambia un sistema de software en forma tal que no altere el comportamiento externo del código, pero sí mejore la estructura interna. Es una manera disciplinada de limpiar el código (y modificar o simplificar el diseño interno) que minimiza la probabilidad de introducir errores. En esencia, cuando se rediseña, se mejora el diseño del código después de haber sido escrito (Pressman, 2010).

Un concepto central en XP es que el diseño ocurre tanto antes como después de que comienza la codificación. Rediseñar significa que el diseño se hace de manera continua conforme se construye el sistema. En realidad, la actividad de construcción en sí misma dará al equipo XP una guía para mejorar el diseño (Pressman, 2010).

Codificación: Después de que las historias han sido desarrolladas y de que se ha hecho el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir en la entrega en curso (incremento de software). Una vez creada la prueba unitaria, el desarrollador está mejor capacitado para centrarse en lo que debe implementarse para pasar la prueba. No se agrega nada extraño (MS). Una vez que el código está terminado, se le aplica de inmediato una prueba unitaria, con lo que se obtiene retroalimentación instantánea para los desarrolladores (Pressman, 2010).

Un concepto clave durante la actividad de codificación (y uno de los aspectos del que más se habla en la XP) es la programación por parejas. XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. Esto da un mecanismo para la solución de problemas en tiempo real (es frecuente que dos cabezas piensen más que una) y para el aseguramiento de la calidad también en tiempo real (el código se revisa conforme se crea). También mantiene a los desarrolladores centrados en el problema de que se trate. En la práctica, cada persona adopta un papel un poco diferente. Por ejemplo, una de ellas tal vez piense en los detalles del código de una porción particular del diseño, mientras la otra se asegura de que se siguen los estándares de codificación (parte necesaria de XP) o de que el código para la historia satisfará

la prueba unitaria desarrollada a fin de validar el código confrontándolo con la historia (Pressman, 2010).

Pruebas: La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas (de modo que puedan ejecutarse en repetidas veces y con facilidad). Esto estimula una estrategia de pruebas de regresión siempre que se modifique el código (lo que ocurre con frecuencia, dada la filosofía del rediseño en XP) (Pressman, 2010).

A medida que se organizan las pruebas unitarias individuales en un “grupo de prueba universal”, las pruebas de la integración y validación del sistema pueden efectuarse a diario. Esto da al equipo XP una indicación continua del avance y también lanza señales de alerta si las cosas marchan mal. Wells dice: “Corregir pequeños problemas cada cierto número de horas toma menos tiempo que resolver problemas enormes justo antes del plazo final” (Pressman, 2010).

Las pruebas de aceptación XP, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software (Pressman, 2010).

3.2.6 El debate XP

Volatilidad de los requerimientos: Como el cliente es un miembro activo del equipo XP, los cambios a los requerimientos se solicitan de manera informal. En consecuencia, el alcance del proyecto cambia y el trabajo inicial tiene que modificarse para dar acomodo a las nuevas necesidades (Pressman, 2010).

Los requerimientos se expresan informalmente. Las historias de usuario y las pruebas de aceptación son la única manifestación explícita de los requerimientos en XP (Pressman, 2010).

Falta de un diseño formal: XP desalienta la necesidad del diseño de la arquitectura y, en muchas instancias, sugiere que el diseño de todas las clases debe ser relativamente informal. Los críticos argumentan que cuando se construyen sistemas complejos, debe ponerse el énfasis en el diseño con el objeto de garantizar que la estructura general del software tendrá calidad y que será susceptible de recibir mantenimiento. Los defensores de XP sugieren que la naturaleza incremental del proceso XP limita la complejidad (la sencillez es un valor fundamental), lo que reduce la necesidad de un diseño extenso (Pressman, 2010).

4. CAPITULO 4: Metodología

Aquí se abordará la metodología XP.

4.1 Iteración 1

4.1.1 Planificación

En esta fase el Cliente coloco las prioridades a cada historia de usuario, que le corresponde, los desarrolladores realizan apreciaciones de lo necesario para realizar cada una de ellas se usara la siguiente plantilla.

Tabla 1 Módulo de Ingreso Sistema

Historia de Usuario	
Número: 1	Usuario: Administrador y Empleado
Nombre historia: Módulo de Ingreso Sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 3	Iteración asignada: 1
Programadores responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere una pantalla para el ingreso al sistema. El usuario requiere que se pida usuario y contraseña. El usuario requiere que se valide los dos campos anteriores y se ingrese.	
Observaciones: El ingreso se permitirá solo a los usuarios registrados en INOCIV.	

Tabla 2 Módulo de Ingreso Sistema-Roles

Historia de Usuario	
Número: 2	Usuario: Administrador
Nombre historia: Módulo de Ingreso Sistema-Roles	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programadores responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere que existan dos tipos de usuarios: administrador y empleados. El usuario requiere que el administrador tenga acceso total a todas las funcionalidades del sistema. El usuario requiere que los empleados tengan acceso a ciertas funcionalidades del sistema.	
Observaciones: - El usuario administrador tiene acceso total del sistema. - El usuario empleado tiene acceso restringido a algunas funcionalidades del sistema.	

4.1.2 Diseño

En esta fase el único entregable que se genera son las tarjetas CRC.

Tabla 3 Usuarios

Clase	
Usuarios	
Responsabilidad	Colaboradores
insertarUsuario() validarSesion() editarUsuario() modificarUsuario() subirArchivo()	RegistrarUsuarioGui.php UsuarioDp.php UsuarioMd.php Funciones.php

4.1.3 Codificación

Para la codificación se utilizaron los estándares de codificación PSR 1 y PSR 2, PSR 0 ya es obsoleto por lo cual no se utiliza en este desarrollo.

Ambos estándares se utilizan como una buena práctica en lo que se refiere a codificación en PHP, lo que ayuda a que el código sea legible y apoye a su mantenimiento, es decir la modificación de cualquier función del software por parte del desarrollador.

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**
- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayusculas de la siguiente forma : **PROYECTOS**
- Nombrado de la primary key de la siguiente forma: **proyectoId**

- Nombrado de los campos de la siguiente forma: **proyectoNombre**

4.1.4 Pruebas

Tabla 4 Prueba Ingreso Sistema

CASO DE PRUEBA	
Numero Caso de Prueba: CP-01	Numero Historia de Usuario: 1
Nombre de Caso de Prueba: Ingreso Sistema	
Descripción: Prueba del módulo de ingreso al sistema.	
Condiciones de ejecución: El usuario tiene que estar registrado en el sistema web, si no lo está no debe ingresar.	
Entradas: Ingresar en los campos de ingreso al sistema el usuario y la contraseña.	
Resultado esperado: Si el usuario esta registrado el ingreso al sistema debe ser satisfactorio, si no está registrado no puede ingresar.	
Evaluación: Prueba exitosa	

Tabla 5 Prueba Ingreso Sistema-Roles

CASO DE PRUEBA	
Numero Caso de Prueba: CP-02	Numero Historia de Usuario: 2
Nombre de Caso de Prueba: Ingreso Sistema-Roles	
Descripción: Prueba del módulo de ingreso al sistema-roles.	
Condiciones de ejecución: Existen dos tipos de usuarios: administrador y empleado, ambos usuarios pueden ingresar al sistema, pero con diferentes funcionalidades.	
Entradas: Ingresar en modo administrador y en modo empleado.	
Resultado esperado: El usuario administrador tiene acceso a todas las funcionalidades del sistema, el usuario empleado tiene acceso a ciertas funcionalidades del sistema.	
Evaluación: Prueba exitosa	

4.2 Iteración 2

4.2.1 Planificación

Tabla 6 Módulo de Pantalla de Inicio

Historia de Usuario	
Número: 3	Usuario: Administrador y Empleado
Nombre historia: Módulo de Pantalla de Inicio	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere una pantalla de inicio. El usuario requiere que en la pantalla de inicio exista una breve descripción de lo que en el sistema se puede realizar. El usuario requiere que se acceda de una manera rápida desde los ítems de la descripción. El usuario requiere que al lado izquierdo también estén los ítems detallados de las funcionalidades del sistema. Las funcionalidades del sistema son: Inicio, Proyectos, Estudios, Registrar Empleados, Registrar Usuarios, Sesiones Usuario, Cerrar Sesión.	
Observaciones: - La pantalla de inicio deberá contener las funcionalidades del sistema con un link que redirige a la funcionalidad seleccionada. - El usuario empleado solo podrá ver en la pantalla de inicio las funcionalidades: Roles de pago, Proyecto, Estudios.	

4.2.2 Diseño

Tabla 7 Bitácoras

Clase	
Bitacoras	
Responsabilidad validarSesion()	Colaboradores BitacoraGui.php UsuarioDp.php UsuarioMd.php

4.2.3 Codificación

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**
- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayusculas de la siguiente forma :
PROYECTOS
- Nombrado de la primary key de la siguiente forma: **proyectoId**
- Nombrado de los campos de la siguiente forma: **proyectoNombre**

4.2.4 Pruebas

Tabla 8 Prueba Pantalla Inicio

CASO DE PRUEBA	
Numero Caso de Prueba: CP-03	Numero Historia de Usuario: 3
Nombre de Caso de Prueba: Pantalla Inicio	
Descripción: Prueba del módulo de pantalla de inicio.	
Condiciones de ejecución: En la pantalla de inicio la descripción de las funcionalidades del administrador deben de ser diferentes a la del empleado.	
Entradas: Ingresar en modo administrador y en modo empleado.	
Resultado esperado: El usuario administrador tiene una pantalla de inicio con acceso a todas las funcionalidades del sistema, el usuario empleado tiene una pantalla de inicio con acceso a ciertas funcionalidades del sistema.	
Evaluación: Prueba favorable	

4.3 Iteración 3

4.3.1 Planificación

Tabla 9 Módulo de Gestión de Proyectos

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre historia: Módulo de Gestión de Proyectos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere ingresar un nuevo proyecto. El usuario requiere que los campos a ingresar del proyecto sean: Nombre, Fecha Inicio/ Fin, Precio, Términos de referencia, Archivo TDR. El usuario requiere una lista de los proyectos que existen en INOCIV. El usuario requiere un encabezado con una breve descripción de cada proyecto. El usuario requiere que los títulos de la descripción sean: Proyecto, Fecha Inicio/ Fin, Precio, Términos de referencia, Descargar TDR, Actividad, Status, Opciones.	
Observaciones: • El usuario empleado solo podrá ver las actividades del proyecto que esta cargo de él, e ingresar el consolidado de la actividad ya realizada. • Los proyectos son los que se van a realizar en concreto.	

Tabla 10 Módulo de Gestión de Proyectos-Encabezado

Historia de Usuario	
Número: 5	Usuario: Administrador
Nombre historia: Módulo de Gestión de Proyectos-Descripción Encabezado	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere que los títulos de la descripción sean: Proyecto, Fecha Inicio/ Fin, Precio, Términos de referencia, Descargar TDR, Actividad, Status, Opciones. El usuario requiere que en la descripción Proyecto vaya el nombre del proyecto. El usuario requiere que en la descripción Fecha Inicio/Fin este la fecha descrita. El usuario requiere que en la descripción Precio conste el valor total del proyecto. El usuario requiere que en la descripción Términos de referencia conste el nombre del proyecto tal y como se envía. El usuario requiere que en la descripción Descargar TDR se pueda descargar el archivo del proyecto. El usuario requiere que en la descripción Actividad se encuentren más opciones sobre ese proyecto. El usuario requiere que en la descripción Status se encuentre un check para ver si está completo. El usuario requiere que en la descripción Opciones se pueda actualizar datos del proyecto.	
Observaciones: El usuario empleado solo podrá ver las actividades del proyecto que esta cargo de él, e ingresar el consolidado de la actividad ya realizada.	

Tabla 11 Módulo de Gestión de Proyectos-Actividad

Historia de Usuario	
Número: 6	Usuario: Administrador y Empleado
Nombre historia: Módulo de Gestión de Proyectos-Descripción Actividad	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere que los campos a ingresar en la nueva actividad sean: Empleado a cargo, Actividad, Descripción. El usuario requiere que al presionar en la descripción Actividad, se pueda ver el título, descripción, status El usuario requiere que exista una parte que se llame Opciones para poder modificar información del proyecto tal como: Descripción, Empleado a cargo y que el status se pueda cambiar cuando todo esté completo. El usuario requiere además subir un archivo que se llamara consolidado.	
Observaciones: El usuario empleado solo podrá ver las actividades del proyecto que esta cargo de él, e ingresar el consolidado de la actividad ya realizada.	

Tabla 12 Módulo de Gestión de Proyectos-Opciones

Historia de Usuario	
Número: 7	Usuario: Administrador y Empleado
Nombre historia: Módulo de Gestión de Proyectos-Descripción Opciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere se pueda modificar información del proyecto tal como: Título, Fechas, Precio y el Archivo. El usuario requiere que se pueda ver y modificar información sobre el consolidado.	
Observaciones: • Los proyectos son los que se van a realizar en concreto. • El usuario empleado solo podrá subir el consolidado con la descripción.	

4.3.2 Diseño

Tabla 13 Proyectos

Clase	
Proyectos	
Responsabilidad insertarProyecto() modificarProyecto() subirArchivo()	Colaboradores ProyectoGui.php ProyectoDp.php ProyectoMd.php Funciones.php

Tabla 14 Actividad Proyectos

Clase	
ActividadProyectos	
Responsabilidad insertarActividadProyecto() modificarActividadProyecto()	Colaboradores ProyectoGui.php ActividadProyectoGui.php ActividadProyectoDp.php ActividadProyectoMd.php

Tabla 15 Proyecto Actividad Consolidados

Clase	
ProyectoActividadConsolidados	
Responsabilidad insertarConsolidadoActividadProyecto() modificarConsolidadoActividadProyecto()	Colaboradores ProyectoGui.php ActividadProyectoGui.php ConsolidadoActividadProyectoGui.php ConsolidadoActividadProyectoDp.php ConsolidadoActividadProyectoMd.php

Tabla 16 Consolidado Proyectos

Clase	
ConsolidadoProyectos	
Responsabilidad insertarConsolidadoProyecto() modificarConsolidadoProyecto() subirArchivo()	Colaboradores ProyectoGui.php ConsolidadoProyectoGui.php ConsolidadoProyectoDp.php ConsolidadoProyectoMd.php

4.3.3 Codificación

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**
- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayúsculas de la siguiente forma :

PROYECTOS

- Nombrado de la primary key de la siguiente forma: **proyectoId**
- Nabrado de los campos de la siguiente forma: **proyectoNombre**

4.3.4 Pruebas

Tabla 17 Prueba Gestión Proyectos

CASO DE PRUEBA	
Numero Caso de Prueba: CP-04	Numero Historia de Usuario: 4
Nombre de Caso de Prueba: Gestión Proyectos	
Descripción: Prueba del módulo de gestión de proyectos.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador e ingresar un nuevo proyecto además de ver la descripción de los proyectos que están ingresados, el usuario empleado no tiene acceso a esta opción.	
Entradas: Ingresar en modo administrador e ingresar un proyecto nuevo y ver los detalles de los demás proyectos ingresados.	
Resultado esperado: Se creará un proyecto nuevo, además se podrá visualizar los detalles de los demás proyectos que están en el sistema.	
Evaluación: Prueba exitosa	

Tabla 18 Prueba Gestión Proyectos-Encabezado

CASO DE PRUEBA	
Numero Caso de Prueba: CP-05	Numero Historia de Usuario: 5
Nombre de Caso de Prueba: Gestión Proyectos-Encabezado	
Descripción: Prueba del módulo de gestión de proyectos-descripción encabezado.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y ver la descripción de todos los proyectos que están ingresados en el sistema, el usuario empleado no tiene acceso a esta opción.	
Entradas: Ingresar en modo administrador y ver los detalles de los demás proyectos ingresados en el sistema.	
Resultado esperado: Se puede ver los detalles de todos los proyectos ingresados en el sistema.	
Evaluación: Prueba exitosa	

Tabla 19 Prueba Gestión Proyectos-Actividad

CASO DE PRUEBA	
Numero Caso de Prueba: CP-06	Numero Historia de Usuario: 6
Nombre de Caso de Prueba: Gestión Proyectos-Actividad	
Descripción: Prueba del módulo de gestión de proyectos-descripción actividad.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador e ingresar una nueva actividad y ver la descripción de todas las actividades que están ingresadas en el sistema, el usuario empleado tiene acceso a esta opción para poder ver la actividad que está asignada a él y además puede subir un archivo.	
Entradas: Ingresar en modo administrador e ingresar una nueva actividad y ver los detalles, ingresar en modo empleado y ver los detalles de la actividad asignada además de subir un archivo.	
Resultado esperado: Se creara una actividad nueva y se podrá visualizar las demás actividades del sistema, el usuario empleado solo podrá ver qué actividad está asignada a él y cargar un archivo con éxito.	
Evaluación: Prueba exitosa	

Tabla 20 Prueba Gestión Proyectos-Opciones

CASO DE PRUEBA	
Numero Caso de Prueba: CP-07	Numero Historia de Usuario: 7
Nombre de Caso de Prueba: Gestión Proyectos-Opciones	
Descripción: Prueba del módulo de gestión de proyectos-descripción opciones.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y modificar la información del proyecto que desee, además se puede modificar información sobre el archivo consolidado.	
Entradas: Ingresar en modo administrador y modificar información del proyecto y del archivo consolidado.	
Resultado esperado: Se modificara la información deseada con éxito.	
Evaluación: Prueba exitosa	

4.4 Iteración 4

4.4.1 Planificación

Tabla 21 Módulo de Gestión de Estudios

Historia de Usuario	
Número: 8	Usuario: Administrador
Nombre historia: Módulo de Gestión de Estudios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere ingresar un nuevo estudio. El usuario requiere que los campos a ingresar del estudio sean: Nombre, Fecha Inicio/ Fin, Precio, Términos de referencia, Archivo TDR. El usuario requiere una lista de los estudios que existen en INOCIV. El usuario requiere un encabezado con una breve descripción de cada estudio. El usuario requiere que los títulos de la descripción sean: Estudio, Fecha Inicio/ Fin, Precio, Términos de referencia, Descargar TDR, Actividad, Status, Opciones.	
Observaciones: • El usuario empleado solo podrá ver las actividades del estudio que esta cargo de él, e ingresar el consolidado de la actividad ya realizada. • Los estudios es una proforma que se está estudiando la factibilidad.	

Tabla 22 Módulo de Gestión de Estudios-Encabezado

Historia de Usuario	
Número: 9	Usuario: Administrador
Nombre historia: Módulo de Gestión de Estudios-Descripción Encabezado	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere que los títulos de la descripción sean: Estudio, Fecha Inicio/ Fin, Precio, Términos de referencia, Descargar TDR, Actividad, Status, Opciones. El usuario requiere que en la descripción Estudio vaya el nombre del estudio. El usuario requiere que en la descripción Fecha Inicio/Fin este la fecha descrita. El usuario requiere que en la descripción Precio conste el valor total del estudio. El usuario requiere que en la descripción Términos de referencia conste el nombre del estudio tal y como se va a enviar. El usuario requiere que en la descripción Descargar TDR se pueda descargar el archivo del estudio. El usuario requiere que en la descripción Actividad se encuentren más opciones sobre ese estudio. El usuario requiere que en la descripción Status se encuentre un check para ver si está completo. El usuario requiere que en la descripción Opciones se pueda actualizar datos del estudio.	
Observaciones: El usuario empleado solo podrá ver las actividades del estudio que esta cargo de él, e ingresar el consolidado de la actividad ya realizada.	

Tabla 23 Módulo de Gestión de Estudios-Actividad

Historia de Usuario	
Número: 10	Usuario: Administrador y Empleado
Nombre historia: Módulo de Gestión de Estudios-Descripción Actividad	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere que los campos a ingresar en la nueva actividad sean: Empleado a cargo, Actividad, Descripción. El usuario requiere que al presionar en la descripción Actividad, se pueda ver el título, descripción, status El usuario requiere que exista una parte que se llame Opciones para poder modificar información del estudio tal como: Descripción, Empleado a cargo y que el status se pueda cambiar cuando todo esté completo. El usuario requiere además subir un archivo que se llamara consolidado.	
Observaciones: El usuario empleado solo podrá ver las actividades del proyecto que esta cargo de él, e ingresar el consolidado de la actividad ya realizada.	

Tabla 24 Módulo de Gestión de Estudios-Opciones

Historia de Usuario	
Número: 11	Usuario: Administrador y Empleado
Nombre historia: Módulo de Gestión de Estudios-Descripción Opciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere se pueda modificar información del estudio tal como: Título, Fechas, Precio y el Archivo. El usuario requiere que se pueda ver y modificar información sobre el consolidado.	
Observaciones: • Los estudios es una proforma que se está estudiando la factibilidad. • El usuario empleado solo podrá subir el consolidado con la descripción.	

4.4.2 Diseño

Tabla 25 Estudios

Clase	
Estudios	
Responsabilidad insertarEstudio () modificarEstudio() subirArchivo()	Colaboradores EstudioGui.php EstudioDp.php EstudioMd.php Funciones.php

Tabla 26 Actividad Estudios

Clase	
ActividadEstudios	
Responsabilidad insertarActividadEstudio () modificarActividadEstudio()	Colaboradores ProyectoGui.php ActividadEstudioGui.php ActividadEstudioDp.php ActividadEstudioMd.php

Tabla 27 Estudio Actividad Consolidados

Clase	
EstudioActividadConsolidados	
Responsabilidad insertarConsolidadoActividadEstudio() modificarConsolidadoActividadEstudio()	Colaboradores ProyectoGui.php ActividadEstudioGui.php ConsolidadoActividadEstudioGui.php ConsolidadoActividadEstudioDp.php ConsolidadoActividadEstudioMd.php

Tabla 28 Consolidado Estudios

Clase	
ConsolidadoEstudios	
Responsabilidad insertarConsolidadoEstudio() modificarConsolidadoEstudio() subirArchivo()	Colaboradores ProyectoGui.php ConsolidadoEstudioGui.php ConsolidadoEstudioDp.php ConsolidadoEstudioMd.php

4.4.3 Codificación

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**

- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayúsculas de la siguiente forma :

PROYECTOS

- Nombrado de la primary key de la siguiente forma: **proyectoId**
- Nombrado de los campos de la siguiente forma: **proyectoNombre**

4.4.4 Pruebas

Tabla 29 Prueba Gestión Estudios

CASO DE PRUEBA	
Numero Caso de Prueba: CP-08	Numero Historia de Usuario: 8
Nombre de Caso de Prueba: Gestión Estudios	
Descripción: Prueba del módulo de gestión de estudios.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador e ingresar un nuevo estudio además de ver la descripción de los estudios que están ingresados, el usuario empleado no tiene acceso a esta opción.	
Entradas: Ingresar en modo administrador e ingresar un estudio nuevo y ver los detalles de los demás estudios ingresados.	
Resultado esperado: Se creará un estudio nuevo, además se podrá visualizar los detalles de los demás estudios que están en el sistema.	
Evaluación: Prueba exitosa	

Tabla 30 Prueba Gestión Estudios-Encabezado

CASO DE PRUEBA	
Numero Caso de Prueba: CP-09	Numero Historia de Usuario: 9
Nombre de Caso de Prueba: Gestión Estudios-Encabezado	
Descripción: Prueba del módulo de gestión de estudios-descripción encabezado.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y ver la descripción de todos los estudios que están ingresados en el sistema, el usuario empleado no tiene acceso a esta opción.	
Entradas: Ingresar en modo administrador y ver los detalles de los demás estudios ingresados en el sistema.	
Resultado esperado: Se puede ver los detalles de todos los estudios ingresados en el sistema.	
Evaluación: Prueba exitosa	

Tabla 31 Prueba Gestión Estudios-Actividad

CASO DE PRUEBA	
Numero Caso de Prueba: CP-10	Numero Historia de Usuario: 10
Nombre de Caso de Prueba: Gestión Estudios-Actividad	
Descripción: Prueba del módulo de gestión de estudios-descripción actividad.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador e ingresar una nueva actividad y ver la descripción de todas las actividades que están ingresadas en el sistema, el usuario empleado tiene acceso a esta opción para poder ver la actividad que está asignada a él y además puede subir un archivo.	
Entradas: Ingresar en modo administrador e ingresar una nueva actividad y ver los detalles, ingresar en modo empleado y ver los detalles de la actividad asignada además de subir un archivo.	
Resultado esperado: Se creara una actividad nueva y se podrá visualizar las demás actividades del sistema, el usuario empleado solo podrá ver qué actividad está asignada a él y cargar un archivo con éxito.	
Evaluación: Prueba exitosa	

Tabla 32 Prueba Gestión Estudios-Opciones

CASO DE PRUEBA	
Numero Caso de Prueba: CP-11	Numero Historia de Usuario: 11
Nombre de Caso de Prueba: Gestión Estudios-Opciones	
Descripción: Prueba del módulo de gestión de estudios-descripción opciones.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y modificar la información del estudio que desee, además se puede modificar información sobre el archivo consolidado.	
Entradas: Ingresar en modo administrador y modificar información del estudio y del archivo consolidado.	
Resultado esperado: Se modificara la información deseada con éxito.	
Evaluación: Prueba exitosa	

4.5 Iteración 5

4.5.1 Planificación

Tabla 33 Módulo de Registro de Empleados-Módulo Roles de Pago

Historia de Usuario	
Número: 12	Usuario: Administrador y Empleado
Nombre historia: Módulo de Registro de Empleados-Modulo Roles de Pago	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 5
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere ingresar un nuevo empleado que trabaja en INOCIV. El usuario requiere que los campos a ingresar del empleado sean: Nombre, Apellido, Teléfono, Dirección, Fecha de ingreso, Salario, Hora de entrada y salida. El usuario requiere una lista de los empleados registrados en INOCIV. El usuario requiere un encabezado con una breve descripción de cada empleado. El usuario requiere que los títulos de la descripción sean: Id, Nombre, Teléfono, Dirección, Fecha de ingreso, Horarios, Salario, Opciones.	
Observaciones: - Solo el usuario administrador tiene acceso a esta opción. - El usuario empleado tiene acceso a esta opción con el nombre de rol de pagos.	

Tabla 34 Módulo de Registro de Empleados-Opciones

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre historia: Módulo de Registro de Empleados-Descripción Opciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 5
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere se pueda modificar información del empleado tal como: Nombre, Apellido, Teléfono, Dirección, Salario. El usuario requiere que se pueda ver y modificar información sobre el Rol de pagos.	
Observaciones: - Solo el usuario administrador tiene acceso a esta opción. - El usuario empleado podrá ver esta opción solo para ver el rol de pagos.	

4.5.2 Diseño

En esta fase el único entregable que se genera son las tarjetas CRC.

Tabla 35 Empleados

Clase	
Empleados	
Responsabilidad insertarEmpleado() editarEmpleado()	Colaboradores RegistrarEmpleadoGui.php EmpleadoMd.php EmpleadoDp.php

4.5.3 Codificación

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**
- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayusculas de la siguiente forma : **PROYECTOS**
- Nombrado de la primary key de la siguiente forma: **proyectoId**
- Nombrado de los campos de la siguiente forma: **proyectoNombre**

4.5.4 Pruebas

Tabla 36 Prueba Registro Empleados-Roles de Pago

CASO DE PRUEBA	
Numero Caso de Prueba: CP-12	Numero Historia de Usuario: 12
Nombre de Caso de Prueba: Registro Empleados-Roles de Pago	
Descripción: Prueba del módulo de registro de empleados-modulo roles de pago.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador e ingresar un nuevo empleado y rol de pago, además de ver la descripción de todos los empleados ingresados, el usuario empleado solo puede ver la descripción del rol de pago y podrá descargarlo .	
Entradas: Ingresar en modo administrador e ingresar un estudio nuevo empleado y rol de pago y ver los detalles de los demás empleados ingresados, ingresar en modo empleado y revisar los detalles de los roles de pagos asignados y descargarlos.	
Resultado esperado: Se ingresara un nuevo empleado y rol de pago, y el usuario empleado vera sus roles de pago y los descargara.	
Evaluación: Prueba exitosa	

Tabla 37 Prueba Registro Empleados-Opciones

CASO DE PRUEBA	
Numero Caso de Prueba: CP-13	Numero Historia de Usuario: 13
Nombre de Caso de Prueba: Registro Empleados-Opciones	
Descripción: Prueba del módulo de registro de empleados-descripción opciones.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y modificar la información del empleado que desee, además se puede modificar información sobre el archivo rol de pagos.	
Entradas: Ingresar en modo administrador y modificar información del empleado y del archivo rol de pagos.	
Resultado esperado: Se modificara la información deseada con éxito.	
Evaluación: Prueba exitosa	

4.6 Iteración 6

4.6.1 Planificación

Tabla 38 Módulo de Registro de Usuarios

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre historia: Módulo de Registro de Usuarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 6
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere ingresar un nuevo usuario de INOCIV. El usuario requiere que los campos a ingresar del empleado sean: Usuario, Nombre, Email, Contraseña, Tipo de usuario, Archivo Foto. El usuario requiere una lista de los usuarios de INOCIV. El usuario requiere un encabezado con una breve descripción de cada usuario. El usuario requiere que los títulos de la descripción sean: Id, Nombre, Contraseña, Email, Fecha Creación, Teléfono, Opciones.	
Observaciones: El usuario administrador podrá crear usuario de tipo: administrador y empleado.	

Tabla 39 Módulo de Registro de Usuarios-Opciones

Historia de Usuario	
Número: 15	Usuario: Administrador
Nombre historia: Módulo de Registro de Usuarios-Descripción Opciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1	Iteración asignada: 6
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere se pueda modificar información del usuario tal como: Usuario, Email, Contraseña. El usuario requiere que se pueda eliminar el usuario.	
Observaciones: Solo el usuario administrador puede eliminar a otro usuario.	

4.6.2 Diseño

Tabla 40 Rol Pagos

Clase	
RolPagos	
Responsabilidad insertarRolPago() modificarRolPago() subirArchivo()	Colaboradores RolPagoGui.php RolPagoMd.php RolPagoDp.php Funciones.php

4.6.3 Codificación

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**
- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayusculas de la siguiente forma : **PROYECTOS**
- Nombrado de la primary key de la siguiente forma: **proyectoId**
- Nabrado de los campos de la siguiente forma: **proyectoNombre**

4.6.4 Pruebas

Tabla 41 Prueba Registro Usuarios

CASO DE PRUEBA	
Numero Caso de Prueba: CP-14	Numero Historia de Usuario: 14
Nombre de Caso de Prueba: Registro Usuarios	
Descripción: Prueba del módulo de registro de usuarios.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador e ingresar un nuevo usuario y foto, además de ver la descripción de todos los usuarios ingresados.	
Entradas: Ingresar en modo administrador e ingresar un estudio nuevo usuario y foto, además de ver los detalles de los demás usuarios ingresados.	
Resultado esperado: Se ingresara un nuevo usuario y foto de forma exitosa.	
Evaluación: Prueba exitosa	

Tabla 42 Prueba Registro Usuarios-Opciones

CASO DE PRUEBA	
Numero Caso de Prueba: CP-15	Numero Historia de Usuario: 15
Nombre de Caso de Prueba: Registro Usuario-Opciones	
Descripción: Prueba del módulo de registro de usuarios-descripción opciones.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y modificar la información del empleado que desee, además se puede eliminar el usuario deseado.	
Entradas: Ingresar en modo administrador y modificar información del empleado y eliminar un usuario.	
Resultado esperado: Se modificara la información deseada y se eliminara el usuario con éxito.	
Evaluación: Prueba exitosa	

4.7 Iteración 7

4.7.1 Planificación

Tabla 43 Módulo de Sesiones de Usuario

Historia de Usuario	
Número: 16	Usuario: Administrador
Nombre historia: Módulo de Sesiones de Usuarios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 7
Programador responsable: Juan Carlos Espinel y Jimmy Navarro	
Descripción: El usuario requiere una lista de los usuarios que han ingresado al sistema. El usuario requiere un encabezado con una breve descripción de cada usuario que ha ingresado. El usuario requiere que los títulos de la descripción sean: Usuario, Descripción, Fecha de sesión.	
Observaciones: Todos los usuarios que ingresen al sistema se podrán ver desde esta funcionalidad.	

4.7.2 Diseño

Tabla 44 Manejo Usuarios

Clase	
Usuarios	
Responsabilidad	Colaboradores
insertarUsuario() validarSesion() editarUsuario() modificarUsuario() subirArchivo()	RegistrarUsuarioGui.php UsuarioDp.php UsuarioMd.php Funciones.php

4.7.3 Codificación

- En el caso de las funciones y métodos se las puede nombrar de la siguiente manera: **insertarProyecto()**
- Las clases se nombran de la siguiente forma: **ActividadProyecto**
- Las constantes se nombran con mayúsculas
- Las variables comenzaran con minúsculas o de la forma camelCase: **nombreProyecto** o **nombre**
- Todo archivo debe contener su extensión **.php**
- El código debe ser legible, esto quiere decir que no tiene que sobrepasar la longitud de la línea, se recomienda que el estilo no debe pasar de las 120 columnas.
- Tabular es una mala práctica, para lo cual se utiliza en vez de ello, 4 espacios para hacer indentación.

En cuanto a la base de datos se utilizó la nomenclatura convencional, de esta forma:

- Nombrado de una tabla, todo en mayusculas de la siguiente forma : **PROYECTOS**
- Nombrado de la primary key de la siguiente forma: **proyectoId**
- Nabrado de los campos de la siguiente forma: **proyectoNombre**

4.7.4 Pruebas

Tabla 45 Prueba Sesiones Usuarios

CASO DE PRUEBA	
Numero Caso de Prueba: CP-16	Numero Historia de Usuario: 16
Nombre de Caso de Prueba: Sesiones Usuarios	
Descripción: Prueba del módulo de sesiones de usuarios.	
Condiciones de ejecución: Se puede acceder desde el usuario administrador y ver la descripción de todas las sesiones de actividad en el sistema.	
Entradas: Ingresar en modo administrador e ver los detalles de las sesiones.	
Resultado esperado: Se podrá visualizar las sesiones del sistema.	
Evaluación: Prueba exitosa	

5. CAPITULO 5: Documentación

5.1 Manual De Usuario

El manual de usuario tiene como finalidad dar una ilustración gráfica de todas las funcionalidades que posee el sistema, dando así una facilidad al momento de utilizar el sistema para que sea de entera satisfacción y comprensión.

Para ver el manual de usuario detallado véase Anexo 1: Manual de usuario

5.2 Manual Técnico

El manual técnico tiene como finalidad explicar que tecnología utiliza el software, además de dar una explicación de los requerimientos para su correcto funcionamiento y la instalación de todos los componentes necesarios para su correcto funcionamiento.

Para ver el manual de usuario detallado véase Anexo 2: Manual Técnico

6. CAPITULO 6: Conclusiones y Recomendaciones

6.1 Conclusiones

- El plan de disertación realizado fue posible por una dificultad poco común que es llevar todo en archiveros, tomando en cuenta con la tecnología disponible para todos, se llevó a cabo la realización de este tema de disertación para automatizar un inconveniente y que sea más fácil de llevar los proyectos de Ingeniería Civil.
- Una de las formas de llevar el archivero era en hojas de cálculo, pero se vuelve poco óptimo el momento de realizar búsquedas o tener todo lo necesario de una forma estructurada y ordenada, lo cual se puede optimizar realizando un sistema personalizado.
- Fue fundamental saber resolver un problema de una forma adecuada aplicando los conocimientos aprendidos en la carrera, un sistema no siempre resuelve todos los inconvenientes porque puede ser muy amplio y poco requerido, para brindar una buena solución es indispensable adentrarse en las dificultades.
- Es importante decir que se realizó un sistema web por la alta disponibilidad que se va a tener en un futuro de este sistema, actualmente, se lo utilizará de forma local.
- Las herramientas que se utilizaron fueron PHP como lenguaje de programación y MySQL como gestor de bases de datos, seleccionadas para que su total funcionamiento sea en la web.

6.2 Recomendaciones

- Saber diferenciar cuándo es bueno realizar un sistema de escritorio o un sistema web, todo se basa en los requerimientos del usuario y un buen criterio antes de dar la mejor solución.
- Se recomienda que se continúe con la implementación de más módulos si es necesario en el caso de que los conflictos aumenten, uno de los cuales sería, el módulo de contabilidad, dando así una solución más amplia.
- Prestar total énfasis que cuando se realiza un software, la ética tiene que estar por encima de todo, para no perjudicar a nadie por encima de la ley, en el caso de esta plan de disertación fue un caso fundamental que los roles de pago sean transparentes para el empleado tanto en la fecha que se los genera, así como su visualización completa.
- Debería existir en la Escuela de Sistemas, cursos y seminarios de las nuevas tecnologías de programación y las últimas tendencias al momento de desarrollar software.

7. Bibliografía

- Álvarez, A., de las Heredas, R., & Lasa, C. (2012). *Métodos ágiles y Scrum*. Madrid: ANAYA MULTIMEDIA.
- Campderrich Falgueras, B. (2003). *Ingeniería del software*. Barcelona: Editorial UOC.
- Cornejo, A. (2012). *Manual de HTML*. feedbooks.
- Date, C. (2001). *Introducción a los sistemas de bases de datos*. Mexico: Pearson Eucación.
- Galiano, J. M. (2016). *IMPLANTAR SCRUM CON ÉXITO*. Barcelona: UOC.
- Gallego Sánchez, A. (s.f.). *Laravel 5*.
- Ibañez, L. H. (2014). *Base de datos: grado superior*. Madrid: RA-MA Editorial.
- Jiménez, J. Z. (2013). *Aplicaciones Web*. Madrid: Macmillan Iberia, S.A.
- Llarena, E., & Pavón, J. (2016). *Creación de un sitio web con PHP y MySQL*. Madrid: Ra-ma.
- Luján, J. (2016). *HTML5, CSS Y JAVASCRIPT*. Madrid: Alfaomega.
- Marques, M. (2009). *BASES DE DATOS*.
- Mateu, C. (2004). *Desarrollo de aplicaciones web*. Barcelona: Eureka Media.
- Millet, P. B., & de la Iglesia, D. G. (2010). *Laboratorio de PHP y MySQL*. Barcelona: UOC.
- MySQL. (2013). *MySQL 5.0 Reference Manual*.
- Sommerville, I. (2005). *Ingeniería del Software*. Madrid: PEARSON.
- Vázquez Mariño, C. (2008). *Programación en PHP5. Nivel Básico*. Ferrol.

8. Anexos

8.1 Anexo 1: Manual De Usuario

8.1.1 Ingreso al Sistema

Paso 1: Ingresar a la siguiente dirección en el navegador <http://localhost/INOCIV/gui/index.php>, y escribir su usuario y contraseña dependiendo el rol que este asignado (En este caso el administrador)

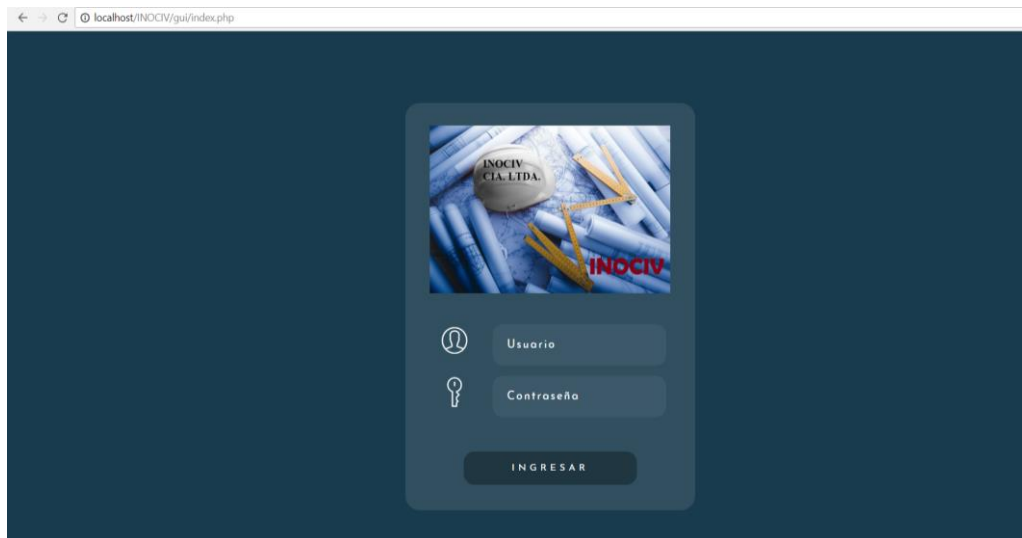


Imagen 1 Anexo 2

Paso 2: Si se autentico correctamente se mostrara la siguiente pantalla, con todas las funciones del sistema.

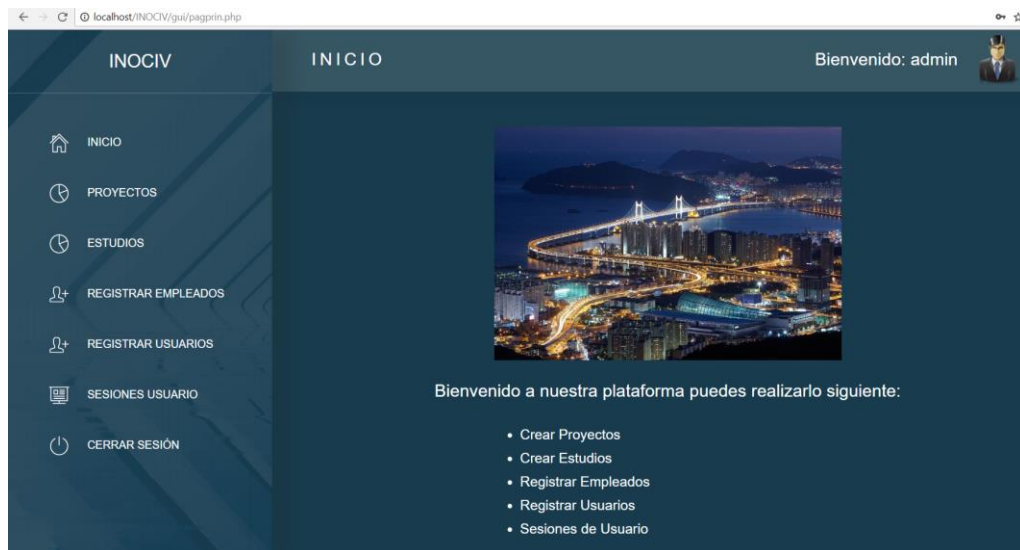


Imagen 2 Anexo 2

8.1.2 Proyectos

Paso 3: Escoger la opción Proyectos y se verá una pantalla de la siguiente manera.



Imagen 3 Anexo 2

Paso 4: Para ingresar un nuevo proyecto dar clic en Nuevo Proyecto.



Imagen 4 Anexo 2

Paso 5: Para descargar un TDR dar clic en Descargar TDR.



Imagen 5 Anexo 2

Paso 6: Para ver la actividad de dicho proyecto, ver los detalles y hacer algunas modificaciones dar clic en Actividad del proyecto que se desee.



Imagen 6 Anexo 2



Imagen 7 Anexo 2

Paso 7: Para modificar datos del proyecto dar clic en Configuración y se tiene dos opciones una de modificar datos y la otra de ver estado del consolidado.



Imagen 8 Anexo 2

8.1.3 Estudios

Paso 8: Escoger la opción Estudios y se verá una pantalla de la siguiente manera.



Imagen 9 Anexo 2

Paso 9: Para ingresa un nuevo estudio dar clic en Nuevo Estudio.



Imagen 10 Anexo 2

Paso 10: Para descargar un TDR dar clic en Descargar TDR.



Imagen 11 Anexo 2

Paso 11: Para ver la actividad de dicho estudio, ver los detalles y hacer algunas modificaciones dar clic en Actividad del estudio que se desee.



Imagen 12 Anexo 2



Imagen 13 Anexo 2

Paso 12: Para modificar datos del estudio dar clic en Configuración y se tiene dos opciones una de modificar datos y la otra de ver estado del consolidado.



Imagen 14 Anexo 2

8.1.4 Registrar Empleados

Paso 13: Escoger la opción Registrar Empleados y se verá una pantalla de la siguiente manera.



Imagen 15 Anexo 2

Paso 14: Para ingresa un nuevo empleado dar clic en Nuevo Empleado.

The screenshot shows the INOCIV web application interface. On the left is a sidebar menu with options: INICIO, PROYECTOS, ESTUDIOS, REGISTRAR EMPLEADOS, REGISTRAR USUARIOS, SESIONES USUARIO, and CERRAR SESIÓN. The main header includes 'INOCIV', 'INICIO', and 'Bienvenido: admin' with a user profile icon. The main content area is titled 'REGISTRO DE EMPLEADOS' and features a 'NUEVO EMPLEADO' button. A modal form titled 'REGISTRO EMPLEADO' is open, containing the following fields: Nombre (Ingrese Nombre), Apellido (Ingrese Apellido), Telefono (Ingrese Telefono), Dirección (Ingrese Dirección), Fecha Ingreso Empleado (dd/mm/aaaa), Salario (Ingrese Salario), Hora de Entrada (Ingrese Hora 08:00:00), and Hora Salida (Ingrese Hora 16:30:00). At the bottom of the modal are 'Ingresar' and 'Cerrar' buttons. In the background, a table of employees is partially visible.

Id	Nombre	Telefono	Dirección	Fecha Ingreso	Horario Entrada	Horario Salida	Salario	Configuración
1	Gustavo Sanchez	091253352	Av. Eloy Alfaro y Av. Rio Amazonas	2018-08-01 00:00:00	08:00:00	18:00:00	800.00	[Icon]
2	Juan Carlos Espinel	0979195080	Av. La Gasca y Pablo Palacios	2018-08-01 00:00:00	08:00:00	17:00:00	700.00	[Icon]

Imagen 16 Anexo 2

Paso 15: Para modificar datos del empleado dar clic en Configuración y se tiene dos opciones una de modificar datos y la otra de ver rol de pago.

The screenshot shows the INOCIV web application interface. The main content area is titled 'REGISTRO DE EMPLEADOS' and features a 'NUEVO EMPLEADO' button. Below it is a table of employees. Each row in the table has a 'Configuración' icon (a pencil) in the last column, which allows for modifying the employee's data or viewing their payment role.

Id	Nombre	Telefono	Dirección	Fecha Ingreso	Horario Entrada	Horario Salida	Salario	Configuración
1	Gustavo Sanchez	091253352	Av. Eloy Alfaro y Av. Rio Amazonas	2018-08-01 00:00:00	08:00:00	18:00:00	800.00	[Icon]
2	Juan Carlos Espinel	0979195080	Av. La Gasca y Pablo Palacios	2018-08-01 00:00:00	08:00:00	17:00:00	700.00	[Icon]

Imagen 17 Anexo 2

8.1.5 Registrar Usuarios

Paso 16: Escoger la opción Registrar Usuarios y se verá una pantalla de la siguiente manera.



Imagen 18 Anexo 2

Paso 17: Para ingresa un nuevo usuario dar clic en Nuevo Usuario.



Imagen 19 Anexo 2

Paso 18: Para modificar datos del empleado dar clic en Configuración y se tiene dos opciones una de modificar datos y la otra de eliminar usuario.



Imagen 20 Anexo 2

8.1.6 Sesiones Usuario

Paso 19: Escoger la opción Sesiones Usuario y se verá una pantalla de la siguiente manera, aquí aparecen todas la sesiones iniciadas en el sistema.



Imagen 21 Anexo 2

8.1.7 Cerrar Sesión

Paso 20: Escoger la opción Cerrar Sesión si lo que quiere es salir del sistema.



Imagen 22 Anexo 2

8.2 Anexo 2: Manual Técnico

8.2.1 Especificaciones Técnicas

Sistema Operativo: Windows 7 o superior/ Linux

Sistema de Gestión de Base de Datos: MySQL

Lenguaje de Programación: PHP 5, HTML 5, CSS3

Servidor de Aplicaciones: Apache 2.4

Espacio en el Disco: 10 Gb en Disco Duro

Espacio en la memoria: 2Gb RAM mínimo

Extras: Conexión a internet

8.2.2 Instalación De Componentes

Paso 1: Descargar XAMPP de la página oficial.

<https://www.apachefriends.org/index.html>

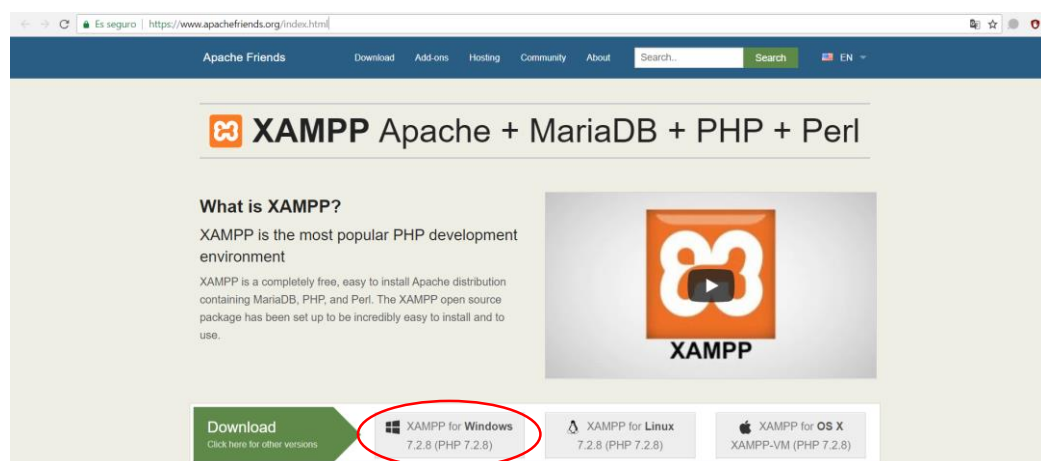


Imagen 1 Anexo 3

Paso 2: Abrir e instalar el archivo descargado.

 xampp-win32-7.2.7-0-VC15-installer.exe	Aplicación	126.824 KB
--	------------	------------

Imagen 2 Anexo 3

Paso 3: Presionar en Next.

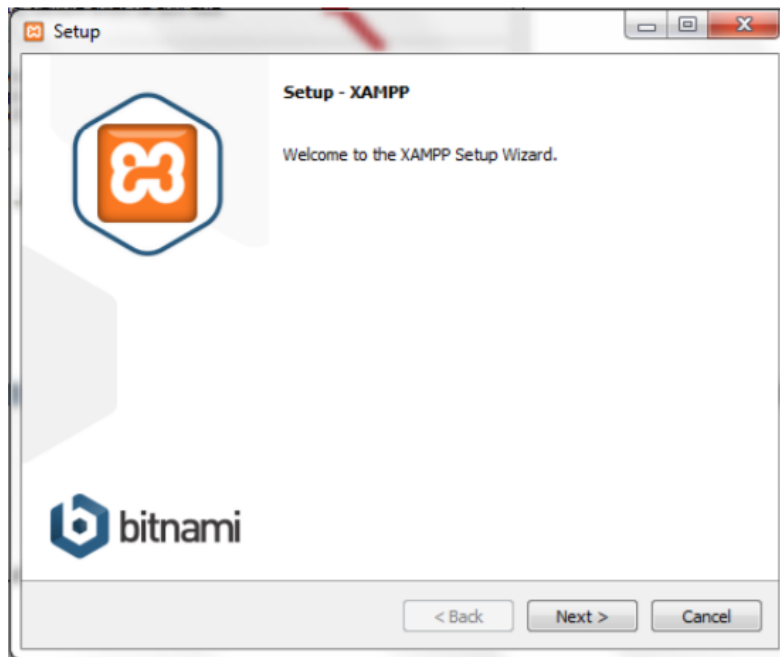


Imagen 3 Anexo 3

Paso 4: Presionar en Next.

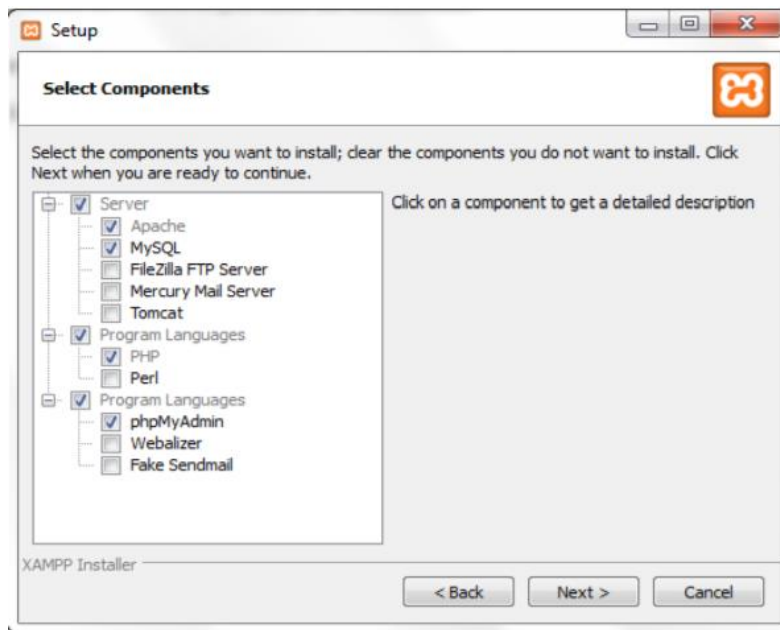


Imagen 4 Anexo 3

Paso 5: Por ultimo presionar en Next para finalizar la instalación.

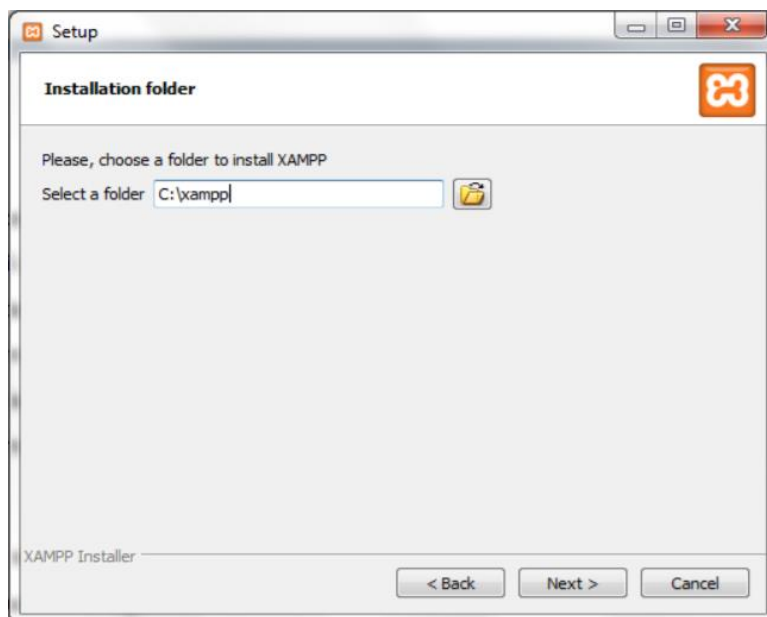


Imagen 5 Anexo 3

Paso 6: Para inicializar XAMPP dar doble clic sobre el icono de la aplicación que se encuentra en el escritorio.



Imagen 6 Anexo 3

La pantalla que se aparecerá será:

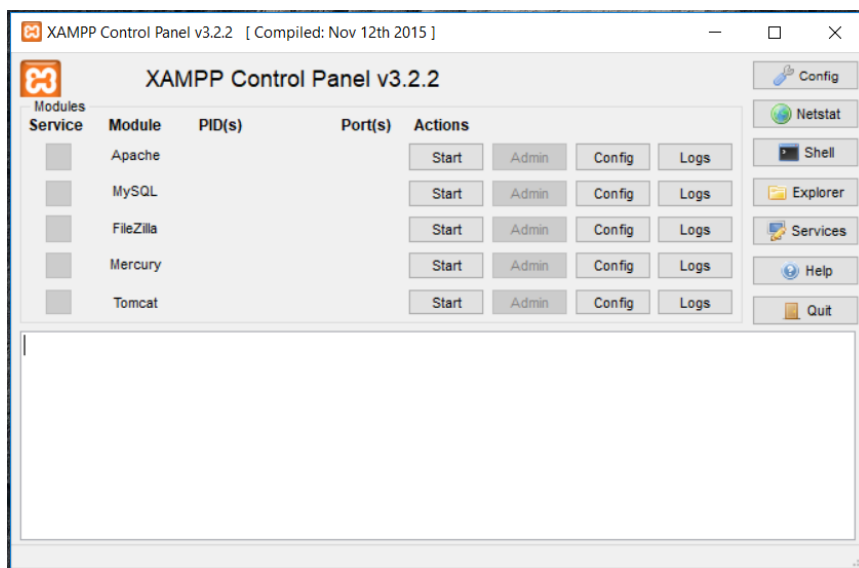


Imagen 7 Anexo 3

Paso 7: Una vez abierto XAMPP damos clic en START para Apache y en START para MySQL.

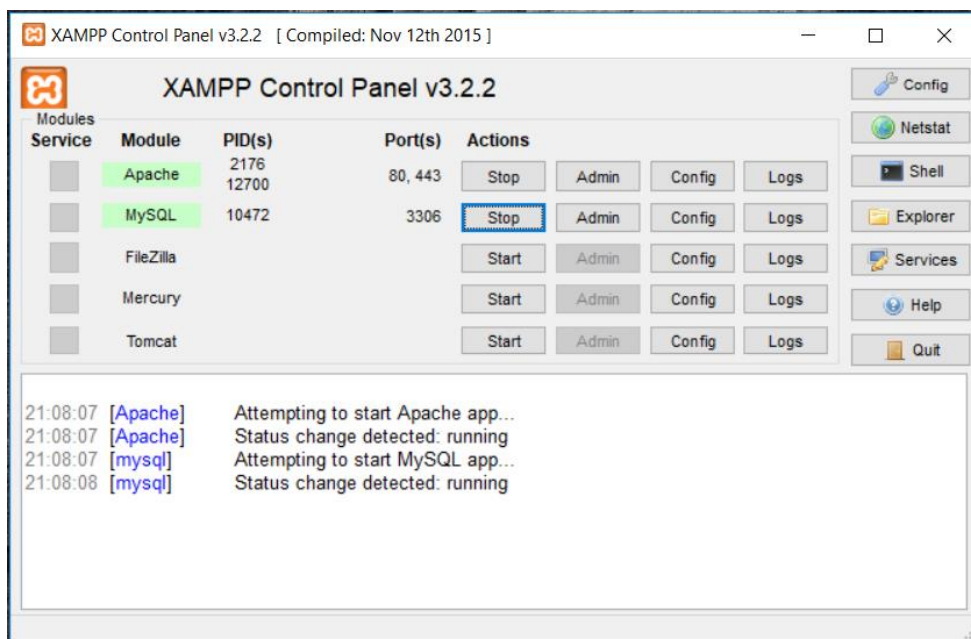
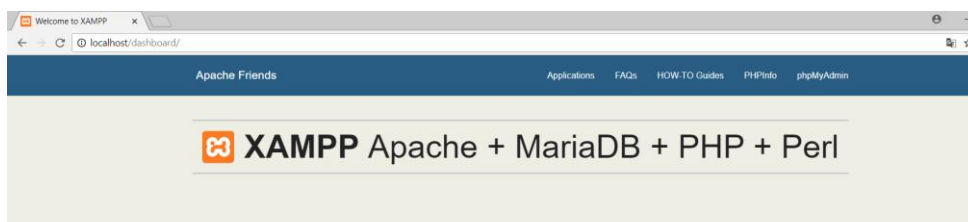


Imagen 8 Anexo 3

Se comprueba el correcto funcionamiento de ambos servicios en estado running.

Paso 8: Ingresar a cualquier navegador y escribir localhost.



Welcome to XAMPP for Windows 7.2.7

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want to have your XAMPP accessible from the internet, make sure you understand the implications and you checked the FAQs to learn how to protect your site. Alternatively you can use WAMP, MAMP or LAMP which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years - there is a huge community behind it. You can get involved by joining our Forums, adding yourself to the Mailing List, and liking us on Facebook, following our exploits on Twitter, or adding us to your Google+ circles.

Contribute to XAMPP translation at translate.apachefriends.org.

Imagen 9 Anexo 3

De esta forma se comprueba que está funcionando correctamente el servicio Apache.

Paso 9: Escribir en el navegador localhost/phpmyadmin.

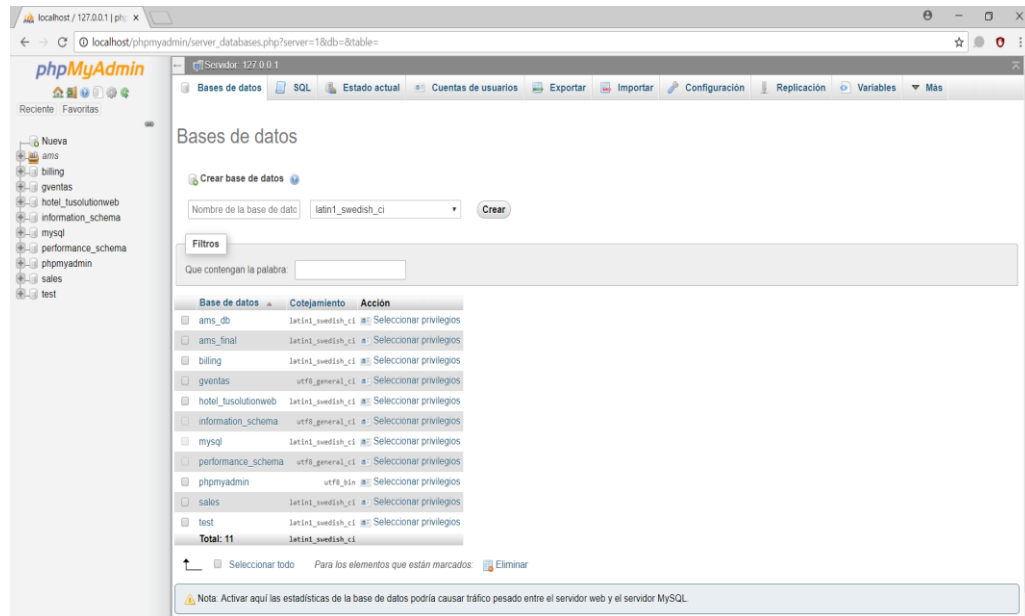


Imagen 10 Anexo 3

De esta forma se comprueba que está funcionando correctamente el servicio MySQL.

Paso 10: Abrir la siguiente carpeta.

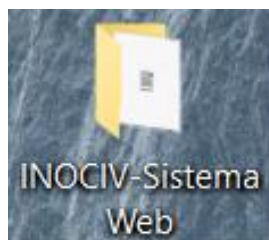


Imagen 11 Anexo 3

Paso 11: Comprobar que se encuentren 1 archivo llamado inocivdb.sql y una carpeta llamada INOCIV.

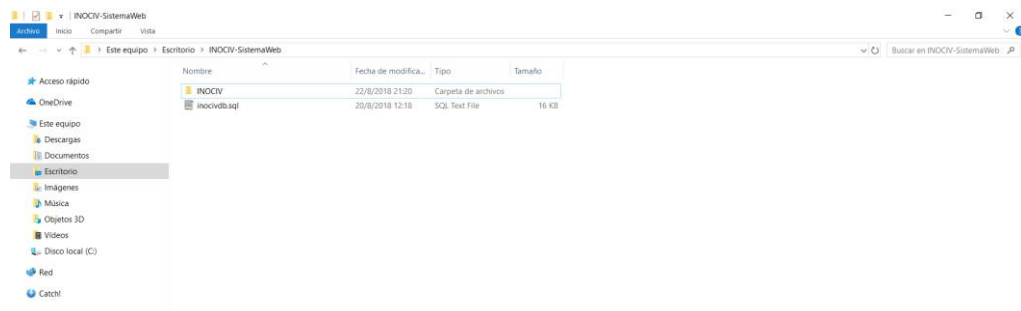


Imagen 12 Anexo 3

Paso 12: Abrir la siguiente dirección en el navegador localhost/phpmyadmin.

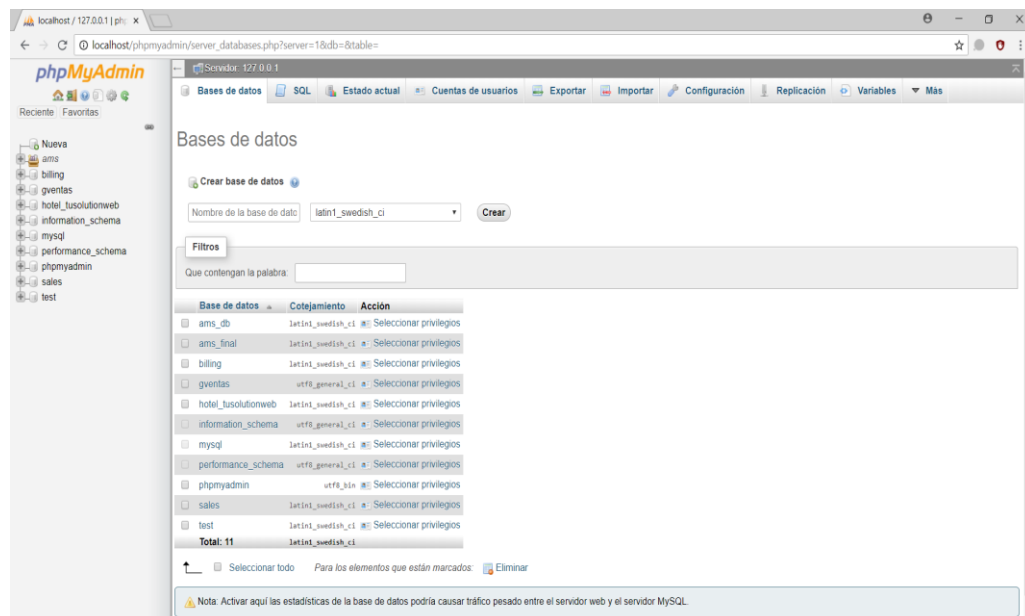


Imagen 13 Anexo 3

Paso 13: Presionar en importar.

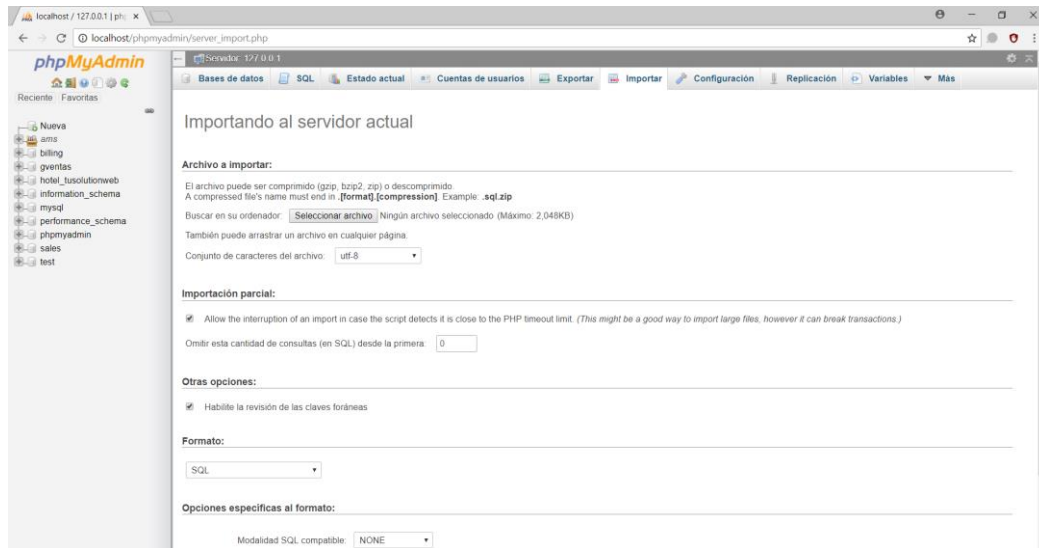


Imagen 14 Anexo 3

Paso 14: Presionar en Seleccionar archivo.

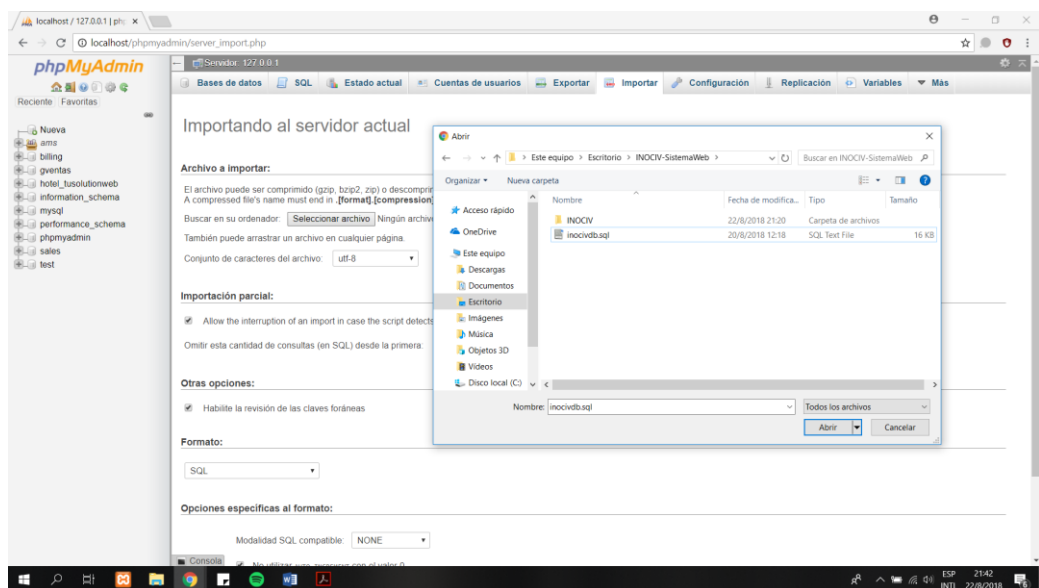


Imagen 15 Anexo 3

Buscar y seleccionar el archivo inocivdb.sql y dar clic en Abrir.

Paso 15: Presionar en Continuar.

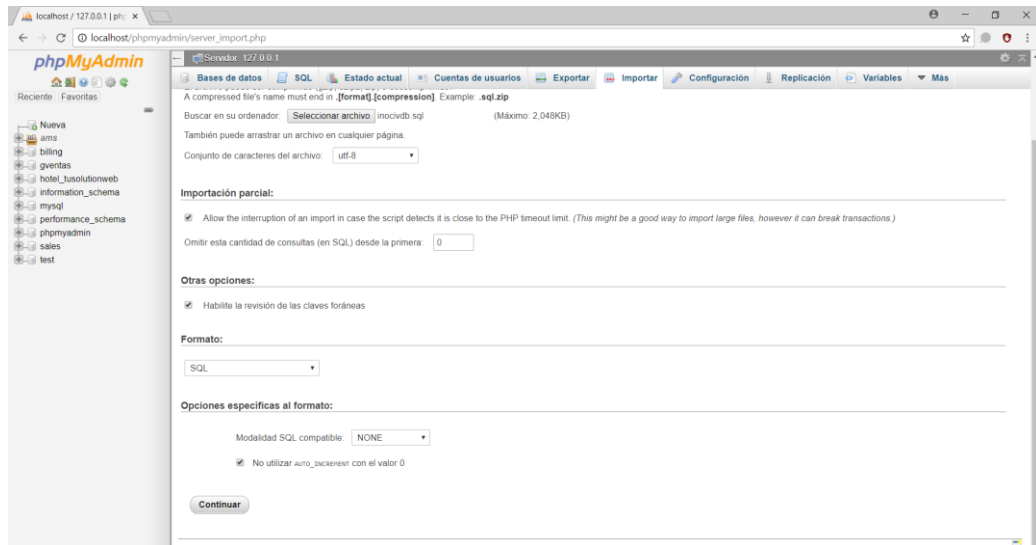


Imagen 16 Anexo 3

Paso 16: Comprobar que todo se haya ejecutado correctamente, aparecerá un aviso de esta forma.

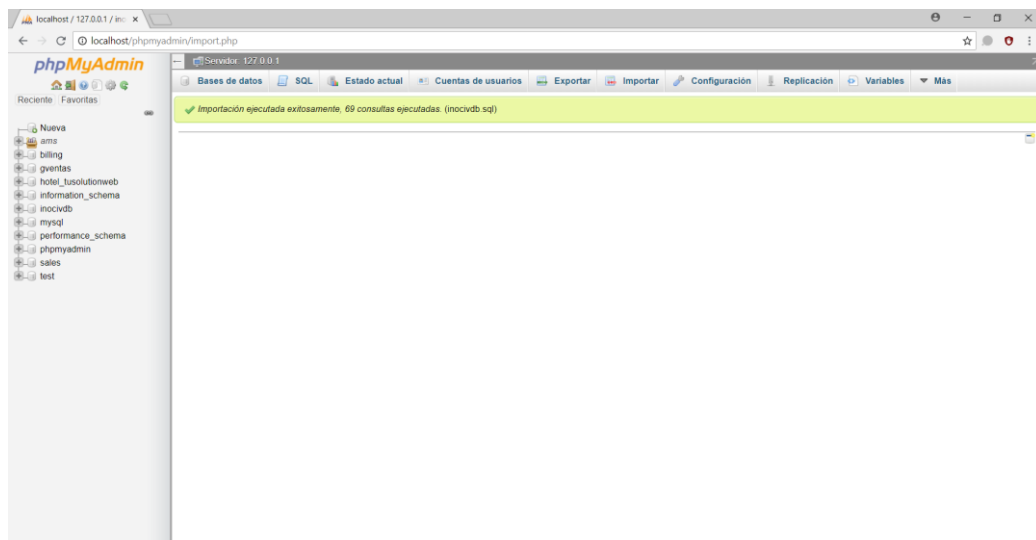


Imagen 17 Anexo 3

Paso 17: Cortar y pegar la carpeta INOCIV en la siguiente ruta C:\xampp\htdocs.

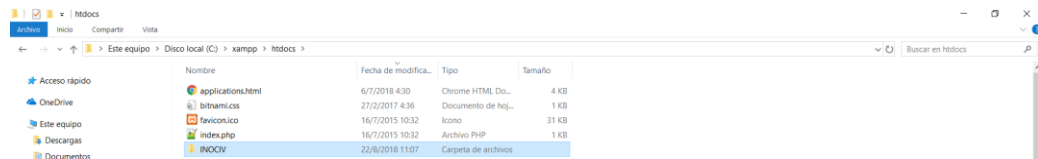


Imagen 18 Anexo 3

Paso 18: Abrir la siguiente dirección en el navegador localhost/INOCIV.

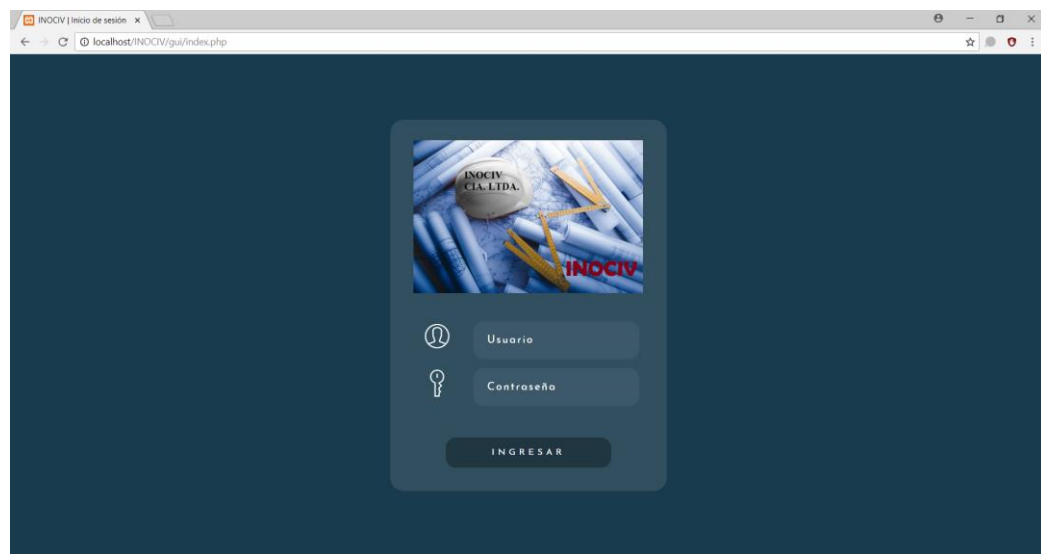


Imagen 19 Anexo 3

Se comprueba que el sistema tenga la pantalla descrita y estará listo para su funcionamiento.