**INDICE DE CONTENIDO**

Contenido

[CAPITULO 1 5](#_Toc529131583)

[SISTEMA DE SEGUIMIENTO DE PASANTIA 5](#_Toc529131584)

[1. INTRODUCCIÓN: 5](#_Toc529131585)

[1.2 DESCRIPCIÓN DEL PROBLEMA 5](#_Toc529131586)

[1.3 OBJETIVO 6](#_Toc529131587)

[1.3.1 Objetivo General 6](#_Toc529131588)

[1.3.2 Objetivo Especifico 6](#_Toc529131589)

[1.4 LIMITES 6](#_Toc529131590)

[CAPITULO 2 8](#_Toc529131591)

[ANTECEDENTES DE LA INSTITUCION 8](#_Toc529131592)

[2. INTRODUCCIÓN 8](#_Toc529131593)

[2.1 MISIÓN 8](#_Toc529131594)

[2.2 VISIÓN 9](#_Toc529131595)

[2.3 PASANTIAS: 9](#_Toc529131596)

[2.3.1 TIPOS DE PASANTÍAS 9](#_Toc529131597)

[2.4 ESPECIALIDAD DE PASANTIAS EN LAS CARRERAS 9](#_Toc529131598)

[2.5 ÁREAS INVOLUCRADAS EN LAS PASANTIAS 13](#_Toc529131599)

[2.5.1 Dirección Académica 13](#_Toc529131600)

[2.5.2 Registros: 13](#_Toc529131601)

[2.5.3 Jefaturas de Carreras: 13](#_Toc529131602)

[2.5.4 Tutores: 13](#_Toc529131603)

[2.5.5 Estudiante: 13](#_Toc529131604)

[2.6 REQUISITOS: 13](#_Toc529131605)

[2.6.1 Requisitos para la Toma de Pasantía 13](#_Toc529131606)

[2.7.2 Requisitos para la Entrega de Documentos de Estudiantes al Tutor. 14](#_Toc529131607)

[2.7.3 Requisitos para la Entrega de Documentos del Tutor a la Unidad de Registros. 15](#_Toc529131608)

[CAPITULO 3 16](#_Toc529131609)

[MARCO TEÓRICO 16](#_Toc529131610)

[3 INTRODUCCIÓN 16](#_Toc529131611)

[3.1 CICLO DE VIDA 16](#_Toc529131612)

[3.2 ELEMENTOS DEL CICLO DE VIDA 17](#_Toc529131613)

[3.2.1 PROCESO BÁSICO DEL CICLO DE VIDA DE UN SISTEMA 17](#_Toc529131614)

[3.2.2 PARADIGMAS DE LOS MODELOS DEL CICLO DE VIDA DEL SOFTWARE 20](#_Toc529131615)

[3.3 DESARROLLO ITERATIVO E INCREMENTAL 21](#_Toc529131616)

[3.3.1 CARACTERÍSTICAS DEL MODELO INCREMENTAL 22](#_Toc529131617)

[3.3.2 VENTAJAS 23](#_Toc529131618)

[3.3.3 INCONVENIENTES 23](#_Toc529131619)

[3.4 METODOLOGIA SCRUM 24](#_Toc529131620)

[3.5 EL PROCESO 24](#_Toc529131621)

[3.6 ROLES DE SCRUM 24](#_Toc529131622)

[3.6.1 PRODUCT OWNER 26](#_Toc529131623)

[3.6.2 SCRUM MASTER 26](#_Toc529131624)

[3.6.2.1 EL SERVICIO DEL SCRUM MASTER AL DUEÑO AL DUEÑO DE PRODUCTO. 26](#_Toc529131625)

[3.6.2.2 EL SERVICIO DEL SCRUM MASTER AL EQUIPO DE DESARROLLO. 27](#_Toc529131626)

[3.6.2.3 EL SERVICIO DEL SCRUM A LA ORGANIZACION 27](#_Toc529131627)

[3.6.3 DEVELOPMENT TEAM. 27](#_Toc529131628)

[3.7 ACTIVIDADES Y ARTEFACTOS 28](#_Toc529131629)

[3.8 HISTORIAS DE USUARIO 28](#_Toc529131630)

[Formato de las historias de usuarios. 29](#_Toc529131631)

[3.9 HERRAMIENTA DE MODELAMIENTO DE LAS FASES DE ANALISIS Y DISEÑO DEL PROYECTO: Lenguaje Unificado de Modelado – UML 30](#_Toc529131632)

[El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación. 30](#_Toc529131633)

[3.9.1 Diagrama de Caso de Uso. 30](#_Toc529131634)

[3.9.1.1 Simbolos de Caso de Uso. 31](#_Toc529131635)

[3.9.2.2 Relaciones de Caso de Uso 33](#_Toc529131636)

[3.10.3 DIAGRAMA DE CLASES 33](#_Toc529131637)

[3.10.3.1 Componentes Basicos de Diagrama de Clases 34](#_Toc529131638)

[3.10.3.2 Interacciones 35](#_Toc529131639)

[3.10.4 DIAGRAMA DE COMPONENTES 36](#_Toc529131640)

[3.10.4.1 descripción 36](#_Toc529131641)

[3.10.4.2 Notación 37](#_Toc529131642)

[3.10.4.3 Componentes 37](#_Toc529131643)

[3.10.4.4 Interfaz 37](#_Toc529131644)

[Ejemplo 38](#_Toc529131645)

[3.10.5 DIAGRAMA DE DESPLIEGUE 39](#_Toc529131646)

[3.10.5.1 Características 39](#_Toc529131647)

[3.10.6 HERRAMIENTAS DE DESARROLLO 41](#_Toc529131648)

[3.10.7 HERRAMIENTAS DE DISEÑO Y DESARROLLO 43](#_Toc529131649)

[3.10.7.1 Visio 2013: 43](#_Toc529131650)

[3.10.7.2 Sqlyog: 43](#_Toc529131651)

[3.10.7.3 MYSQL Workbench: Es un software creado por la empresa Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL. 44](#_Toc529131652)

[3.10.7.4 Sublime Text: 44](#_Toc529131653)

[3.10.7.5 Adobe Photoshop CS6; 45](#_Toc529131654)

[3.10.7.6 Navegador Google Chrome; 45](#_Toc529131655)

[3.10.7.7 Phpstorm Ide. - 46](#_Toc529131656)

[3.10.7.8 Dia. - 46](#_Toc529131657)

[CAPITULO 4 47](#_Toc529131658)

[FASE DE ANALISIS Y DISEÑO 47](#_Toc529131659)

[4 INTRODUCCIÓN: 47](#_Toc529131660)

[4.1 HISTORIAS DE USUARIO DEL PROYECTO 47](#_Toc529131661)

[4.2 MODELO DE CASOS DE USO 50](#_Toc529131662)

[4.2.1 IDENTIFICACIÓN DE ACTORES 51](#_Toc529131663)

[4.2.2 DESCRIPCIÓN CASO DE USO: A continuación, se describirán cada uno de los escenarios más importantes del sistema Internship. 54](#_Toc529131664)

[4.3.2 .1 Escenario 1: Reportes de Pasantías 54](#_Toc529131665)

[4.3.2.2 Escenario 2: Reporte de Estudiante 55](#_Toc529131666)

[4.3.2.3 Escenario 3: Reporte de Jefe de Carrera 56](#_Toc529131667)

[4.3.2.4 Escenario 4: Reporte de Registro 57](#_Toc529131668)

[4.3.2.6 Escenario 6: Revisión documentos Área de Registro 58](#_Toc529131669)

[4.3.2.7 Escenario 7: Seguimiento Tutor al Pasante 58](#_Toc529131670)

[4.3 DIAGRAMA DE CLASES 59](#_Toc529131671)

[4.3.2 ULTIMA VERSIÓN DEL DIAGRAMA DE CLASES: 60](#_Toc529131672)

[4.3.3 DIAGRAMA ENTIDAD RELACIÓN. 61](#_Toc529131673)

[CAPITULO 5 62](#_Toc529131674)

[FASE DE IMPLEMENTACIÓN 62](#_Toc529131675)

[5.1 INTRODUCCIÓN. - 62](#_Toc529131676)

[5.2 ITERACIONES DEL SISTEMA CONTROL Y SEGUIMIENTOS DE PASANTIA 62](#_Toc529131677)

[5.2.1 ITERACIÓN 1 63](#_Toc529131678)

[5.2.1.1 Iteraciones hasta la primera propuesta de desarrollo y apariencia: 63](#_Toc529131679)

[PRIMERA VERSION 64](file:///C:\Users\PERSONAL\Desktop\PASANTIAS-14-documentacion\SISTEMA%20DE%20SEGUIMIENTO%20DE%20PASANTIA%20Revision13.docx#_Toc529131680)

[5.2.2 ITERACIÓN 2.- 65](#_Toc529131681)

[5.2.2.1 Iteraciones hasta la segunda propuesta de desarrollo del diseño. 66](#_Toc529131682)

[5.2.3 ITERACIÓN 3.- 71](#_Toc529131683)

[A. Logueo del sistema Control y Seguimiento de Pasantías. – 72](#_Toc529131684)

[5.3 DIAGRAMA DE COMPONENTES. – 76](#_Toc529131685)

[5.4 DIAGRAMA DE DESPLIEGUE. - 77](#_Toc529131686)

[5.5 INTERFAZ DE SISTEMA 78](#_Toc529131687)

[A. Administrador 78](#_Toc529131688)

[CAPITULO 6 83](#_Toc529131689)

[CONCLUSIONES 83](#_Toc529131690)

[INTRODUCCION 83](#_Toc529131691)

[CONCLUSIONES ACERCA DE LOS OBJETIVOS 83](#_Toc529131692)

[1. CONCLUSIONES ACERCA DE LAS HERRAMIENTAS UTILIZADAS 83](#_Toc529131693)

[2. CONCLUSIONES ACERCA DE LA METODOLOGÍA 84](#_Toc529131694)

[3. CONCLUSIONES ACERCA DE LA IMPLEMENTACION 84](#_Toc529131695)

[4. RECOMENDACIONES: 85](#_Toc529131696)

[REQUERIMIENTOS DE HARDWARE Y SOFTWARE. 85](#_Toc529131697)

[ANEXO 87](#_Toc529131698)

[DICCIONARIO DE DATOS. 87](#_Toc529131699)

[GLOSARIO DE TERMINOS 100](#_Toc529131700)

[DOCUMENTACION DE PASANTIAS 102](#_Toc529131701)

[Certificado de pasantía. 102](#_Toc529131702)

[CROQUIS DE EMPRESA 103](#_Toc529131703)

[FICHA DE PLAN DE TRABAJO CARRERA 103](#_Toc529131704)

[CONTROL DE EVALUACION: 104](#_Toc529131705)

[FICHA DE REGISTRO DE ASISTENCIA DE PASANTIA 104](#_Toc529131706)

[CRONOGRAMA INTERNO DE SEGUIMIENTO DE PASANTIA 105](#_Toc529131707)

# CAPITULO 1

# SISTEMA DE SEGUIMIENTO DE PASANTIA

# **INTRODUCCIÓN**:

Hoy en día los problemas relacionados con control y seguimiento de la documentación de pasantías, con la mayoría de las carreras de nuestra institución, han causado conflictos en los trámites y solicitud de información de estudiantes egresados o en proceso de pasantía. Algunos problemas son la presentación de documentación de pasantías, pérdida de documentación de pasantías realizadas durante el periodo de la carrera, errores en la documentación presentada entre otros.

Las innovaciones tecnológicas, la mayor expectativa de vida, el nivel de información de la sociedad en su conjunto, y por lo tanto la mayor exigencia, justifican la elaboración de una herramienta para el manejo de información que respalde la documentación manejada en las pasantías.

# 1.2 DESCRIPCIÓN DEL PROBLEMA

Después de haber realizado un análisis de la toma de datos con cuestionarios y entrevistas al personal de la institución involucrado con las pasantías, se pudo evidenciar los siguientes problemas.

La supervisión de la pasantía no lleva un seguimiento formal, en ocasiones el docente o tutor no procede con las inspecciones que se deberían realizar en la empresa que está realizando el pasante.

* No existe evidencia de la documentación presentada del pasante al tutor.
* Existen errores en algunos documentos de pasantías.
* Los estudiantes requieren más información sobre los requisitos para realizar la pasantía.
* No hay modo de verificar que tutor haga las visitas a la institución donde estudiante realiza las pasantías.
* No hay control de las revisiones de cuadernillos de las pasantías.
* En el transcurso de los años se va perdiendo o extraviando la documentación entregada, por el pasante o tutor debido a la mala administración.

# 1.3 OBJETIVO

## 1.3.1 Objetivo General

Desarrollar un sistema de gestión para el control y seguimiento de pasantías.

## 1.3.2 Objetivo Especifico

* Control de Usuarios, mediante uso de privilegios en la gestión de usuarios autorizados.
* Realizar reportes, Constancias y respaldos de la información requerida.
* Realizar un módulo para Respaldo de información mediante una base de datos dedicada a la aplicación.
* Control y seguimiento para el docente o tutor de la pasantía con estudiantes designados en una gestión académica.
* Control y seguimiento de la documentación presentada del tutor de la pasantía.
* Búsqueda de información relacionada a las pasantías para los estudiantes.
* Asignar usuarios y contraseñas.
* Generar reportes específicos de usuarios.

# 1.4 LIMITES

* No realizara la recolección de firmas de la dirección académica y/o encargados del área.
* No reemplazara el rol del docente al momento de la visita a la institución en la cual se realiza la pasantía.
* El módulo de visitas del tutor, requerirá siempre de un móvil o dispositivo electrónico conectado a internet para realizar la evaluación.



**fig. 1** cronograma de actividades

**(Ref.** Elaboración Propia)

# CAPITULO 2

# ANTECEDENTES DE LA INSTITUCION

1. INTRODUCCIÓN

La Fundación INFOCAL Cochabamba creada el año 1989 y a partir de 1996 mediante D.S. 24240 convertida en una Fundación Nacional, es una institución sin fines de lucro, con personalidad jurídica propia, autonomía administrativa, económica y funcionamiento descentralizado a través de Fundaciones Departamentales (regionales), cuyo financiamiento está basado en aportes empresariales destinados a la administración, al desarrollo de los programas de formación y capacitación, así como contraparte a proyectos de cooperación internacional.

El objetivo principal de la Fundación, es desarrollar la formación profesional técnica y la capacitación en diversas actividades productivas y de servicio, en cumplimiento de las exigencias del mercado.

Este accionar, ha sido reconocido con distinciones y menciones, como la de la Asamblea Legislativa Plurinacional de Bolivia (Cámara de Diputados) y la del Concejo Municipal de Cochabamba, nombrando a INFOCAL Cochabamba institución meritoria.

# 2.1 MISI**ÓN**

Es una fundación educativa especializada en la formación de profesionales técnicos y a la capacitación continua. Brinda una educación de alta calidad a nuestros estudiantes con el fin de formar emprendedores con sólidos conocimientos técnicos que cumplan con las necesidades del sector industrial y servicios a través de una educación técnica y práctica, para así aportar al desarrollo social y económico de la sociedad.

# 2.2 VISI**ÓN**

“Ser líder nacional en enseñanza técnica y especializada formando profesionales con sentido humano, emprendedores y competitivos”.

Ser un modelo de excelencia académica en la formación de profesionales, promoviendo un servicio innovador, que inserte a los jóvenes al desarrollo de Bolivia.

# 2.3 PASANTIAS:

Las pasantías son materias regulares de los últimos 2 años de formación que se cursan en todas las carreras de Infocal.

## **TIPOS DE** PASANTÍAS

**2.3.1.1 Anual:**

Involucran realizar 240 horas de trabajo en una empresa sea privada o pública. Realizando 2 pasantías desde segundo hasta tercer año de la carrera, sumados en total 480 horas.

* + - 1. **Semestral:**

Involucran realizar 160 horas de trabajo en una empresa sea privada o pública. Realizando 3 pasantías desde cuarto hasta sexto semestre, sumados en total 480 horas.

# 2.4 ESPECIALIDAD DE PASANTIAS EN LAS CARRERAS

**2.4.1 Gastronomía:**

Actualmente la carrera de gastronomía tiene 2 planes de estudio vigente, anual y semestral. Dentro del régimen semestral se divide en 3 pasantías:

* La primera pasantía de repostería actividades del rubro gastronómico como son: pastelería, panadería, bar, servicio de comedores y otros.
* La segunda pasantía de panadería se organiza, dirige y supervisa las tareas específicas del servicio gastronómico en empresas de eventos y catering.
* La tercera pasantía de cocina se elabora menús nutritivos, saludables y creativos, considerando requerimientos de los diversos clientes.

Y en el régimen anualizado se dividen en 2 pasantías:

* la primera pasantía relacionada a cocina nacional o internacional.
* la segunda pasantía a repostería y panadería.

**2.4.2 Instalaciones Integrales y de gas. -** Dentro del régimen semestral se divide en 3 pasantías:

* La primera pasantía es de Instalaciones de sistemas de fontanería, refrigeración, aire acondicionado y calefacción.
* La segunda pasantía es de Instalaciones domiciliarias y comerciales de gas natural y gas licuado de petróleo.
* La tercera pasantía es de mantenimiento de equipos de gas natural, GLP o eléctricos (calefones, termo tanques, calefactores y otros).

**2.4.3 Mecánica Automotriz:** Dentro del régimen semestral se divide en 3 pasantías que se realizan en un taller que tenga los requisitos requeridos.

* La primera pasantía Inyección gasolina, diésel del automóvil.
* La segunda pasantía electrónica y electricidad del automóvil, gestión de almacenes y control de calidad al servicio técnico.
* La tercera pasantía diagnóstico y reparación de sistemas mecánicos, eléctricos, electrónicos, transmisiones automáticas con sistemas electrónicos, neumáticos e hidráulicos, motores con inyección electrónica a diésel, gasolina, con sistema de Common Rail y GNV.

**2.4.4 Mecánica Industrial:** Dentro del régimen semestral se divide en 3 pasantíasAplicación de procesos tecnológicos en la fabricación de elementos de máquinas y estructuras

* La primera pasantía es de Instalación, montaje, operación, mantenimiento y reparación de máquinas, equipos mecánicos, hidráulicos y neumáticos.
* La segunda pasantía consiste en aplicación de procesos tecnológicos en la fabricación de elementos de máquinas y estructuras.
* La tercera pasantía es de planificación, organización y control para asegurar una eficiente producción industrial.

**2.4.5 Electrotecnia Industrial:** Dentro del régimen semestral se divide en 3 pasantíasAplicación de elementos eléctricos y componentes electrónicos para generar procesos de automatización.

* La primera pasantía es de Interpretación y elaboración de planos y diagramas eléctricos.
* La segunda pasantía es de Mantenimiento de motores y generadores eléctricos.
* La tercera pasantía consiste en aplicación de elementos eléctricos y componentes electrónicos para generar procesos de automatización.

**2.4.6 Parvulario:** Dentro del régimen semestral se divide en 3 pasantías. Aplica conocimientos de psicología, didáctica, planificación curricular y estrategias del aprendizaje, en sus tareas educativas.

* La primera pasantía es plan de juego, aplica conocimientos de psicología, didáctica, planificación curricular y estrategias del aprendizaje, en sus tareas educativas.
* La segunda pasantía es plan de clase, demuestra creatividad e integra fantasía, imaginación, música, danza y expresión corporal en la estimulación motora y comunicativa del niño.
* La tercera pasantía es plan de tema, planifica diferentes actividades que facilitarán la expresión oral y escrita del niño.

**2.4.7 Turismo:** Dentro del régimen semestral se divide en 3 pasantías que se realizan en Hotelería y Turismo donde pueden realizar sus conocimientos adquiridos.

* La primera pasantía es atención en Recepción de Hotel.
* La segunda pasantía es Cocina y atención al cliente en restaurante.
* La tercera pasantía es agencia de viajes.

**2.4.8 Electromecánica Industrial:** Identificación y aplicación de técnicas de fabricación y tratamiento de materiales.

* La primera pasantía consiste en operación de máquinas, herramientas y mantenimiento electromecánico industrial.
* La segunda pasantía consiste en la identificación y aplicación de técnicas de fabricación y tratamiento de materiales.
* La tercera pasantía consiste en la interpretación, diseño, simulación y montaje de sistemas eléctricos industriales. Colaboración en el montaje, instalación y puesta en marcha de sistemas mecánicos, eléctricos, neumáticos e hidráulicos.

**2.4.9 Sistemas Informáticos:** Actualmente realizan 2 pasantías divididas en 240 horas.

La primera pasantía consiste en realizar el ensamblado, mantenimiento preventivo y correctivo de equipos de computación.

La segunda pasantía consiste en realizar diseño e implementación de redes informáticas desde el diagnóstico, análisis, diseño e implementación. asimismo, consisten en el desarrollo de sistemas de información, software para móviles y juegos.

# **2.5 ÁREAS INVOLUCRADAS EN LAS PASANTIAS**

* + 1. Dirección Académica**:** Es la entidad encargada de hacer cumplir todos los reglamentos internos afines a la realización de pasantías y otros relacionados al área académica a través de las jefaturas de carreras en ambos centros.

Al final de la gestión académica se encarga de visar los informes que pasan los jefes de carrera.

El director académico se encarga de revisar y establecer los puntos concernientes a la realización de convenios con empresas privadas o públicas para las diferentes carreras.

* + 1. Registros: Es la unidad encargada de realizar la revisión de toda la documentación referente a las pasantías que presenta el tutor de una carrera para que posteriormente se archiven esos documentos en el file del estudiante y se presenten los informes correspondientes a dirección académica.
    2. Jefaturas de Carreras:Es la unidad encargada derealizar el seguimiento a los tutores de la pasantía para verificar toda la documentación física y digital referente a las pasantías efectuadas en una gestión académica.
    3. Tutores: Es la unidad encargada de hacer el seguimientoal estudiante, haciendo las visitas a la empresa, revisando el cuaderno de pasantías e informando cualquier eventualidad a jefatura de carrera. finalizando con la evaluación del trabajo del pasante junto al supervisor de la empresa y la presentación de toda la documentación a jefatura de carrera.
    4. Estudiante:Es el actor principal, quien debe inscribirse en la pasantía y asistir regularmente a la empresa en los horarios acordados.

# **2.6 REQUISITOS:**

## 2.6.1 Requisitos para la Toma de Pasantía

**1.-** Pasar a inscribirse a la pasantía I, II, III en la gestión actual.

**2.-** Registrar la empresa a realizar pasantía (tenga NIT)

**3.-** Entregar datos de la empresa:(nombre, rubro,

teléfono, dirección y el nombre del gerente o propietario) solicitar formulario de carta pasantía en fotocopiadora o en la unidad recepción/despacho de tramites del área de registros y certificaciones.

**4.-** Entregado el formulario de carta pasantía por jefatura de carrera, llevar a oficinas de recepción/despacho de trámites. Recoger carta en 48 horas.

**5.-** Pasando las 48 horas, recoger la carta de la unidad de recepción, dicha carta entregar a la empresa. La empresa asignara un supervisor para coordinaran los horarios de trabajo.

**nota:** la pasantía debe realizarse fuera de horarios de clase.

**6.-** Pedir formularios de ficha de solicitud de pasantía y la ficha de plan de trabajo de pasantía (según formato de cada carrera).

**7.-** Realiza la pasantía cuando el supervisor de la empresa y estudiante finalizan horarios de trabajo, documentos presentados. Las horas contaran al con documentos entregados.

**8.-** Hacer pago en cajas el cuaderno de pasantía y recoger su cuaderno de almacenes.

**9.-** Asignación de un tutor al pasante, quien hará seguimiento al trabajo del pasante.

## 2.7.2 Requisitos para la Entrega de Documentos de Estudiantes al Tutor.

Finalizando la pasantía (Concluyendo 160 horas en el semestralizado y 240 horas en el anualizado), presentar al tutor:

* + - 1. **Certificado de Pasantía,** el cual deberá estar en hoja membretada con firma del supervisor y jefe de la empresa, con los sellos respectivos en la última hoja del cuaderno.
      2. **Cuaderno de** **Pasantía**: Donde se describe el trabajo de cada día, que se presenta cada semana para la revisión y sellado por tutor y supervisor de la empresa.
      3. **Informe Técnico**: Solicitar en jefatura de carrera, llenar los siguientes datos:
* Resumen de la Actividades Desarrolladas
* Dificultades y Logros
* Conclusiones
* Recomendaciones

## 2.7.3 Requisitos para la Entrega de Documentos del Tutor a la Unidad de Registros.

**2.7.3.1**  **Certificado de Pasantía**: Debe estar limpio, sin suciedad y sin borrones.

**2.7.3.2**  **Ficha de Evaluación Final**- Documento con las firmas del tutor y supervisor de la empresa a registros.

**2.7.3.3 Informe Técnico Estudiante. -**

**2.7.3.4 Informe Final.** – Descripción del docente con toda la información referente a los pasantes.

# CAPITULO 3

# MARCO TEÓRICO

# 3 INTRODUCCIÓN

Es una de las fases más importantes de un trabajo de investigación, consiste en desarrollar la teoría que va a fundamentar el proyecto con base al planteamiento del problema que se ha realizado. Existen numerosas posibilidades para elaborarlo, la cual depende de la creatividad del investigador. Una vez que se ha seleccionado el tema objeto de estudio y se han formulado las preguntas que guíen la investigación, el siguiente paso consiste en realizar una revisión de la literatura sobre el tema. Esto consiste en buscar las fuentes documentales que permitan detectar, extraer y recopilar la información de interés para construir el marco teórico pertinente al problema de investigación planteado.

# 3.1 CICLO DE VIDA

Todo proyecto de ingeniería tiene unos fines ligados a la obtención de un producto, proceso o servicio que es necesario generar a través de diversas actividades. Algunas de estas actividades pueden agruparse en fases porque globalmente contribuyen a obtener un producto intermedio, necesario para continuar hacia el producto final y facilitar la gestión del proyecto. Al conjunto de las fases empleadas se le denomina “ciclo de vida”.

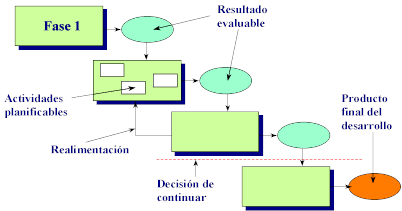
“La complejidad de las relaciones entre las distintas actividades crece exponencialmente con el tamaño, con lo que rápidamente se haría inabordable si no fuera por la vieja táctica de “divide y vencerás”. De esta forma la división de los proyectos en fases sucesivas es un primer paso para la reducción de su complejidad, tratándose de escoger las partes de manera que sus relaciones entre sí sean lo más simples posibles.”[[1]](#footnote-1)

De la misma forma, la práctica acumulada en el diseño de modelos de ciclo de vida para situaciones muy diversas permite que nos beneficiemos de la experiencia adquirida utilizando el enfoque que mejor se adapte a nuestros requerimientos.

# 3.2 ELEMENTOS DEL CICLO DE VIDA

Un ciclo de vida para un proyecto se compone de fases sucesivas compuestas por tareas panificables. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con bucles de realimentación, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones de los resultados intermedios que se van produciendo (realimentación).

**Fig. 2** Ciclo de vida



(**Ref.** [http://www.spw.cl/proyectos/apuntes2/images/cvida1.gif VISITA: 08/SEP/2018)](http://www.spw.cl/proyectos/apuntes2/images/cvida1.gif%20VISITA:%2008/SEP/2018))

## 3.2.1 PROCESO BÁSICO DEL CICLO DE VIDA DE UN SISTEMA

Si bien, es cierto que existen diversas metodologías y formas de desarrollar software, la realidad es que hay modelos tan antiguos que ya son como básicos al momento del ciclo de vida de un software. Un ejemplo de esto, es el modelo en cascada para el proceso de desarrollo de un sistema, con el cuál veremos a ciencia cierta el proceso básico, del cual muchos modelos más se empezarán a desarrollar.

**3.2.1.1** Planificación. El primer punto importante en el ciclo de vida de software, es analizar brevemente los requerimientos que el cliente pide para la elaboración del sistema que necesita. Esta etapa requiere se cierto conocimiento para poder entender la idea que el cliente propone, además de que regularmente debes tomar nota con cada uno de los puntos importantes que se te solicitan, de este modo puedes hacer una planificación al momento y llegar incluso a determinar los tiempos de desarrollo que te llevará, antes de proceder a entregar el producto final. Un punto importante por el cual la planificación siempre debe estar en los ciclos de vida del software. Es porque el cliente se imagina su producto final de una forma tan abstracta, que necesitas hacer que ponga los pies en el suelo para obtener resultados que se acerquen más a la realidad.

**3.2.1.2** Implementación**.** Una vez que hemos platicado con el cliente y tenemos lo que es un análisis de requerimientos, necesidades y funcionalidades por parte de una aceptación en ambas partes, entonces procedemos con lo que es el ciclo de vida de desarrollo de software. Para este punto, existen una infinidad de metodologías de desarrollo de software, que nos ofrecen la posibilidad de trabajar de distintas formas. Más adelante hablaremos más específicamente de ellas, sin embargo, la implementación, es básicamente la parte donde los programadores empiezan a codificar o desarrollar el sistema que se necesita, básicamente se trata del ciclo de vida del desarrollo de sistemas, sin importar el lenguaje de programación mediante el cual se vayan a elaborar.

**3.2.1.3** Pruebas**.** Una vez que el sistema se va desarrollando, es importante para el ciclo de vida del desarrollo del software, que se realicen ciertas pruebas conforme se vaya avanzando. La idea es que no se termine el desarrollo para poder hacer pruebas, si no que mucho antes, durante el proceso de creación, estas ya se puedan ir ejecutando. Las pruebas nos van a permitir ver si el sistema que se está desarrollando es funcional, si tiene algunos errores, si le faltan ciertas cosas para funcionar correctamente, pues básicamente para avanzar al siguiente punto del ciclo de desarrollo de software, será necesario haber pasado las pruebas correctamente.

**3.2.1.4** Documentación**.** Muchas metodologías de lo que es el ciclo de vida software, van creando documentación, conforme se va avanzando en el desarrollo del sistema. Sin embargo, algunas otras prefieren no hacer la documentación hasta el final. Ahora sí que sea cual sea la metodología que elijas, la documentación siempre será importante, pues considera que no siempre vas a estar tú y tu equipo disponibles y cuando otro equipo llegue a programar lo que ustedes hicieron, será indispensable que haya una documentación de la cual se puedan basar, para poder empezar a desarrollar nuevamente el sistema incompleto.

**3.2.1.5** Despliegue**.** Ya casi llegando a lo que son las últimas etapas del desarrollo de software, nos encontramos con el Despliegue. Este no es otra cosa, más que el momento en que el sistema ya está terminado y ha sido aprobado para que se elabore el producto final. Ahora será el momento de distribuirlo y celebrar, pues gracias al equipo de trabajo es como se habrá llegado a esta fase. Lamentablemente, de las etapas de desarrollo de software, esta es a la cual muchos nunca llegan. Pues una gran cantidad de software incompleto se queda en el camino debido a distintos puntos o motivos. Puede ser que el equipo no se unió, el cliente declinó, el proyecto no fue funcional, etc. Así que, de haber llegado a esta fase de desarrollo de software, tu como tu equipo deberá sentirse orgullosos y es momento de volver a desarrollar un proyecto más.

**3.2.1.6** Mantenimiento**.** La última de las fases del desarrollo de software, es el mantenimiento. Que creías, que nunca más verías al software que hicieron, terminaron y distribuyeron. Pues claro que, si lo volverías a ver, pues es momento de darle mantenimiento. Acá además se pueden agregar lo que son las actualizaciones, dependiendo del tipo de desarrollo. Si el equipo siguió trabajando con el software desarrollado y encontraron formas de hacerle mejoras, entonces parte del mantenimiento será actualizarlo a la versión final en todo momento.

## 3.2.2 PARADIGMAS DE LOS MODELOS DEL CICLO DE VIDA DEL SOFTWARE

“Una de las cosas principales, que se deben elegir al momento de empezar un proyecto de desarrollo de software, son precisamente las etapas del desarrollo de software. Si bien, nos queda claro que no todos tenemos las mismas ideas y no todos pensamos de la misma manera, afortunadamente ya existen modelos preestablecidos bajo los cuales podemos elaborar nuestro proyecto. Es por eso que a continuación les mostraré cuales son algunos de los paradigmas de los Modelos del ciclo de vida de desarrollo de sistemas”[[2]](#footnote-2). Bajo los cuales podemos encontrar una gran cantidad de modelos distintos para desarrollar software, veamos.

**3.2.2.1** Paradigma Tradicional.Existen algunas metodologías del ciclo de vida de desarrollo de sistemas, que se manejan a la antigua, a estas también se le conocen como paradigmas tradicionales. Si bien, es verdad que las metodologías actuales están basadas con fundamentos de lo que fueron los paradigmas tradicionales, hoy en día ya hemos evolucionado, sin embargo, los paradigmas tradicionales ahí se mantienen. Estos paradigmas, se caracterizan principalmente por ser lineales sin vuelta atrás, es decir, se trataba de completar cada proceso de principio a fin, hasta que quedara listo para avanzar a la segunda fase del ciclo del software. Esto generaba grandes dificultades y pérdidas de tiempo si se encontraba algún error en una fase avanzada, pues el proceso a realizarse era, volver atrás y volver a pasar nuevamente por las fases que ya se habían hecho y reestructurar de acuerdo a las modificaciones, pero todo con un proceso lineal, lento y tardado.

**3.2.2.2** Paradigma Orientado a Objetos.Una de las genialidades más exquisitas, es el desarrollo de software mediante programación orientada a objetos. Con esta forma del ciclo de vida de los sistemas, lo que se pretende es que el código fuente sea reutilizable para otros proyectos o mini proyectos alternos relacionados con el programa base, pues se utilizan Clases. Básicamente la etapa de desarrollo de software en el paradigma orientado a objetos, se conforma principalmente lo que es la creación de clases, seguido del análisis de requisitos, un paso fundamental para determinar no solamente la duración del desarrollo, sino también los costos al final del proyecto. Y por supuesto el diseño, pues ya con el paradigma orientado a objetos, el diseño es mucho mejor que con un paradigma tradicional.

**3.2.2.3** Paradigma de Desarrollo Ágil**.** Los modelos de ciclo de vida ágiles, son de los más utilizados hoy en día. El objetivo de este paradigma, es el desarrollo de proyectos en poco tiempo. Para lo cual, se hace una eliminación de procesos tediosos, se agilizan las fases de desarrollo, las iteraciones se hacen en un corto periodo de tiempo, los riesgos se desechan y se evitan para no tener que lidiar con ellos y siempre se da solución a los problemas de forma rápida. Si algo demora mucho en dar solución, lo mejor es dejarlo de lado y seguir avanzando. Una de las principales diferencias del paradigma de desarrollo ágil con los paradigmas anteriores, es que el cliente se ve involucrado en el proyecto durante el desarrollo de este. A diferencia del paradigma tradicional donde el cliente solo está al principio, de igual forma en el paradigma orientado a objetos sucede lo mismo. Acá el cliente interfiere, da mejoras, propone ideas y se mantiene al tanto del desarrollo del producto. Lo que ayuda aún más, pues el producto final se realiza de forma satisfactoria en un menor lapso de tiempo.

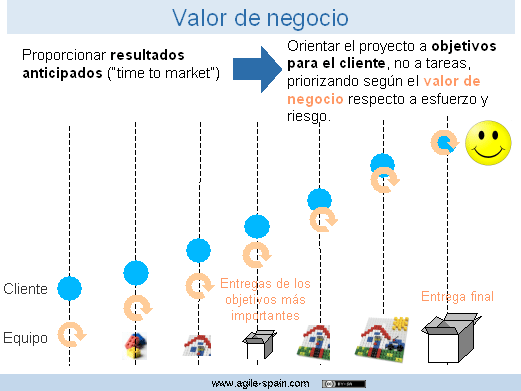
# 3.3 **DESARROLLO ITERATIVO E INCREMENTAL**

En un desarrollo iterativo e incremental el proyecto se planifica en diversos bloques temporales (en el caso de Scrum de un mes natural o hasta de dos semanas, si así se necesita) llamados iteraciones.

“Las iteraciones se pueden entender como mini proyectos: en todas las iteraciones se repite un proceso de trabajo similar (de ahí el nombre “iterativo”) para proporcionar un resultado completo sobre producto final, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental. Para ello, cada requisito se debe completar en una única iteración: el equipo debe realizar todas las tareas necesarias para completarlo (incluyendo pruebas y documentación) y que esté preparado para ser entregado al cliente con el mínimo esfuerzo necesario. De esta manera no se deja para el final del proyecto ninguna actividad arriesgada relacionada con la entrega de requisitos.”[[3]](#footnote-3)

En cada iteración el equipo **evoluciona el producto** (hace una entrega incremental) a partir de los resultados completados en las iteraciones anteriores, añadiendo nuevos objetivos/requisitos o mejorando los que ya fueron completados. Un aspecto fundamental para guiar el desarrollo iterativo e incremental es la [**priorización de los objetivos/requisitos en función del valor que aportan al cliente**](https://proyectosagiles.org/priorizacion-requisitos-valor-coste).

**fig 3**.- Representación de iteraciones



(**Ref.** <https://scrumenespanol.files.wordpress.com/2015/09/valor-negocio-iterativo-incremental.gif?w=521&h=391&zoom=2> VISITA08/sep./2018)

## 3.3.1 CARACTERÍSTICAS DEL MODELO INCREMENTAL

Por el contrario, están vinculadas de forma que cada una suponga un avance con respecto a la anterior. Otras características esenciales de este modelo son:

* Los incrementos son pequeños.
* Permite una fácil administración de las tareas en cada iteración.
* La inversión se materializa a corto plazo.
* Es un modelo propicio a cambios o modificaciones.
* Se adapta a las necesidades que surjan.

Para que esto sea posible, se debe tener en cuenta que las iteraciones no pueden ser demasiado rígidas y que no existan tareas simultáneas. El modelo incremental exige un encadenamiento progresivo de cada tarea. [Scrum](https://www.obs-edu.com/blog-project-management/scrum/scrum-metodologia-el-motivador/)y Kanban son las herramientas más conocidas que emplean este modelo de gestión.

### 3.3.2 VENTAJAS

Entre las ventajas que puede proporcionar un modelo de este tipo encontramos las siguientes:

* Mediante este modelo se genera software operativo de forma rápida y en etapas tempranas del ciclo de vida del software.
* Es un modelo más flexible, por lo que se reduce el coste en el cambio de alcance y requisitos.
* Es más fácil probar y depurar en una iteración más pequeña.
* Es más fácil gestionar riesgos.
* Cada iteración es un hito gestionado fácilmente

### 3.3.3 INCONVENIENTES

Para el uso de este modelo se requiere una experiencia importante para definir los incrementos y distribuir en ellos las tareas de forma proporcionada. Entre los inconvenientes que aparecen en el uso de este modelo podemos destacar los siguientes:

* Cada fase de una iteración es rígida y no se superponen con otras.
* Pueden surgir problemas referidos a la arquitectura del sistema porque no todos los requisitos se han reunido, ya que se supone que todos ellos se han definido al inicio.

# 3.4 METODOLOGIA SCRUM

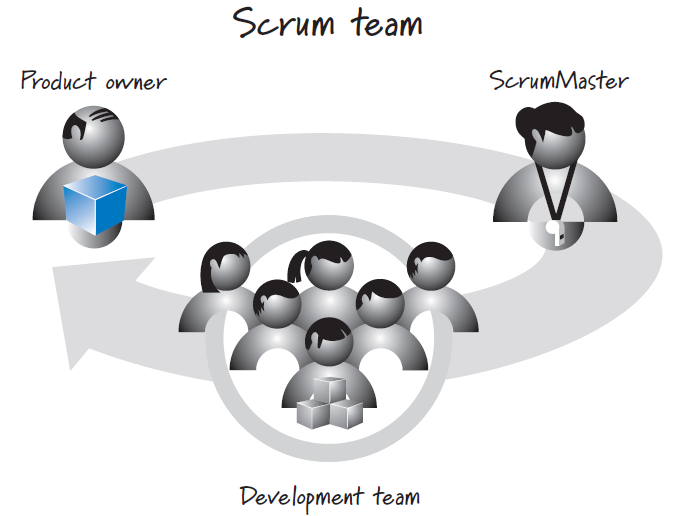
“Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.”

# 3.5 EL PROCESO

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

# 3.6 ROLES DE SCRUM

* Un proyecto de desarrollo se puede llevar a cabo mediante uno o más equipos Scrum
* Un equipo Scrum está formado por personas que juegan tres tipos de roles:
* Product owner
* Scrum Master
* Development team member

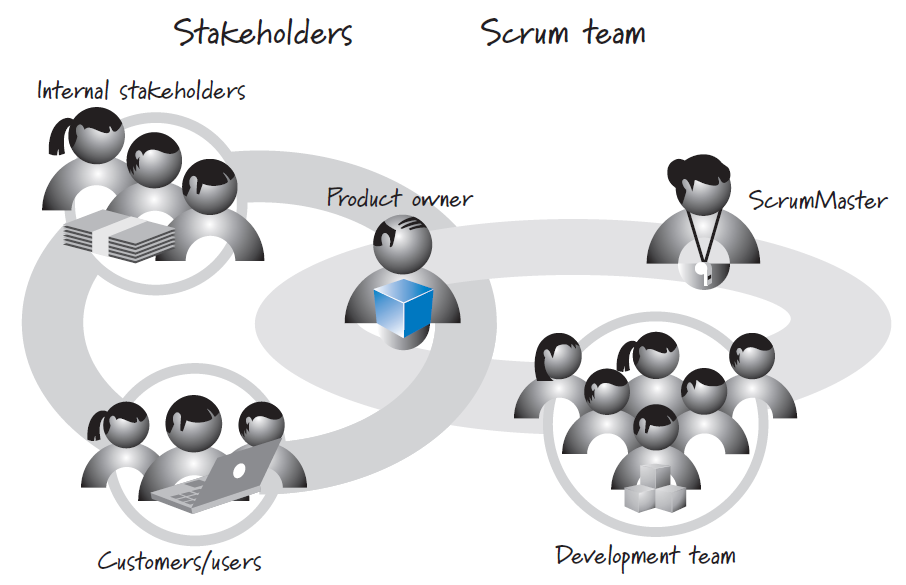
Un equipo Scrum se auto-organiza y no necesita jefes o gestores, aunque si serán necesarios en el contexto de la organización: contratación, formación, establecimiento y control de objetivos, gestión económica, asignación de personas y tareas, etc. (Graf 1)

**Fig**. 4 Ciclo de paso de información a través del Scrum team

(**Ref**. Elaboración Propia)

#### Scrum: Product owner

**Fig**. 5 Product Owner Nexo de conexión entre equipo de desarrollo



(**Ref**. Elaboración Propia)

## 3.6.1 PRODUCT OWNER

“El Product Owner se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner ayuda al usuario a escribir las [historias de usuario](https://es.wikipedia.org/wiki/Historias_de_usuario), las prioriza, y las coloca en el [Product Backlog](https://es.wikipedia.org/w/index.php?title=Scrum_(development)&action=edit&redlink=1).”

* Responsable de la visión de producto y la gestión económica de su desarrollo
* Nexo de conexión entre equipo de desarrollo y stakeholders, clientes y usuarios.
* Decide qué características y funcionalidades ha de tener el sistema en desarrollo y el orden en que deben ser implementadas (Product backlog) y las valida. (Graf 2)
* Participa activamente en el equipo Scrum.

## 3.6.2 SCRUM MASTER

“El Scrum Master es el responsable de asegurar que Scrum se entienda y se adopte (Graf 2). Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El Scrum Master es un líder que está al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser útiles y cuáles no.” [12]

El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

### 3.6.2.1 EL SERVICIO DEL SCRUM MASTER AL DUEÑO AL DUEÑO DE PRODUCTO.

El Scrum Master da servicio al Dueño de Producto de varias formas, incluyendo:

* Encontrar técnicas para gestionar la Lista de Producto de manera efectiva;
* Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos;
* Entender la planificación del producto en un entorno empírico;
* Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor;
* Entender y practicar la agilidad; y,
* Facilitar los eventos de Scrum según se requiera o necesite.

### 3.6.2.2 EL SERVICIO DEL SCRUM MASTER AL EQUIPO DE DESARROLLO.

El Scrum Master da servicio al Equipo de Desarrollo de varias formas, incluyendo:

* Guiar al Equipo de Desarrollo en ser autoorganizado y multifuncional;
* Ayudar al Equipo de Desarrollo a crear productos de alto valor;
* Eliminar impedimentos para el progreso del Equipo de Desarrollo;
* Facilitar los eventos de Scrum según se requiera o necesite; y,
* Guiar al Equipo de Desarrollo en entornos organizacionales en los que Scrum aún no haya sido adoptado y entendido por completo.

### 3.6.2.3 EL SERVICIO DEL SCRUM A LA ORGANIZACION

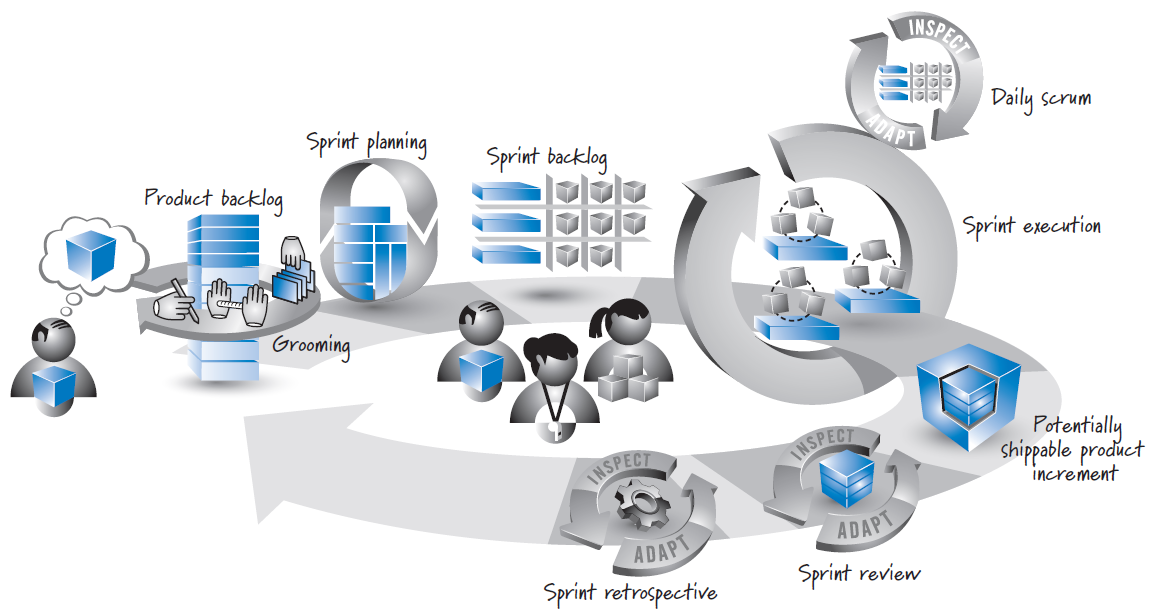
El Scrum Master da servicio a la organización de varias formas, incluyendo:

* Liderar y guiar a la organización en la adopción de Scrum;
* Planificar las implementaciones de Scrum en la organización;
* Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto;
* Motivar cambios que incrementen la productividad del Equipo Scrum; y,
* Trabajar con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

## 3.6.3 DEVELOPMENT TEAM.

* Responsables del diseño, implementación y verificación del sistema en desarrollo
* El equipo se auto-organiza para llevar a cabo los objetivos fijados por el Product owner
* En conjunto, deben tener todos los conocimientos y capacidades para producir software funcional de buena calidad
* Tamaño típico: 5-9 personas. En proyectos mayores se pueden usar jerarquías de equipos.

# 3.7 ACTIVIDADES Y ARTEFACTOS



**Fig**. 6 Iteración Scrum

**(Ref.** Elaboración Propia)

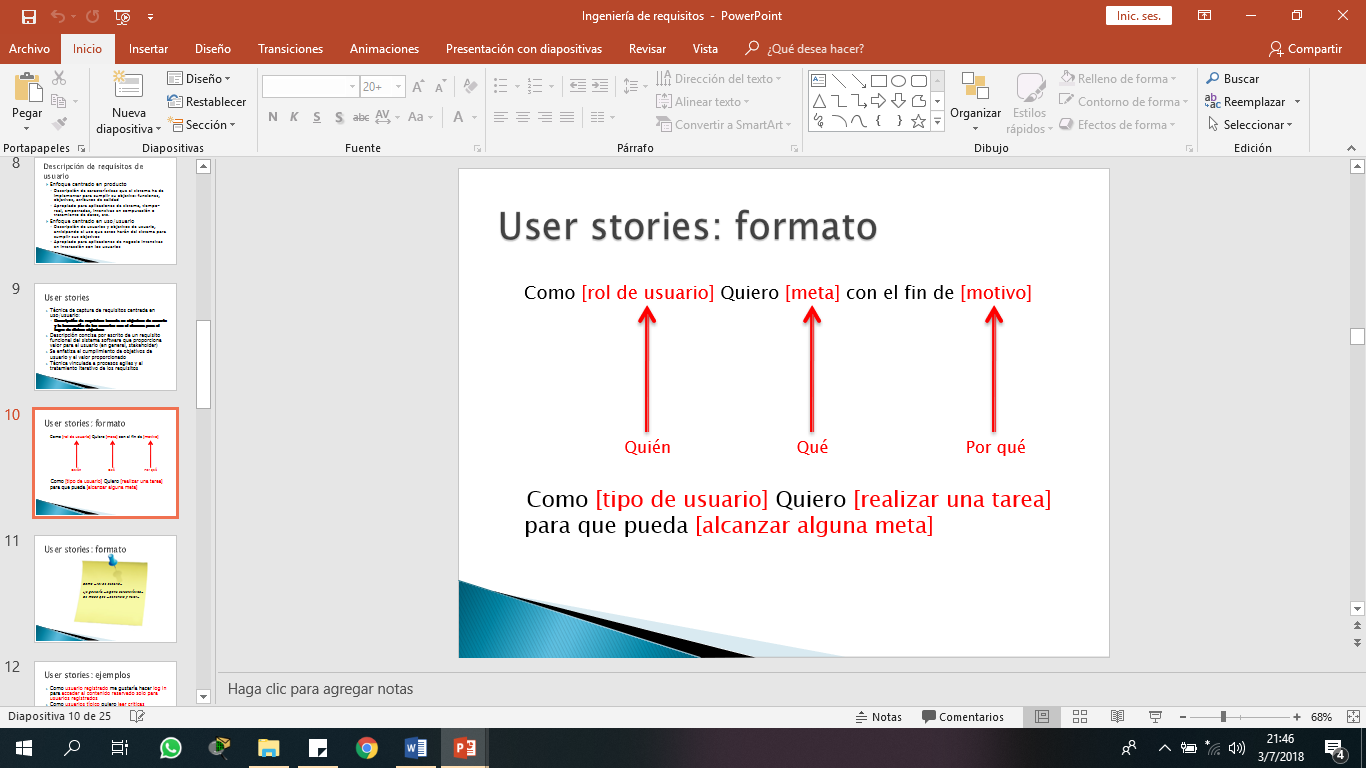
# 3.8 HISTORIAS DE USUARIO

La historia de usuario describe funcionalidad que será útil para el usuario, o comprador de un sistema de software.

* Técnica de captura de requisitos centrada en uso/usuario:
  + Descripción de requisitos basada en objetivos de usuario y la interacción de los usuarios con el sistema para el logro de dichos objetivos
* Descripción concisa por escrito de un requisito funcional del sistema software que proporciona valor para el usuario (en general, stakeholder)
* Se enfatiza el cumplimiento de objetivos de usuario y el valor proporcionado
* Técnica vinculada a procesos agiles y al tratamiento iterativo de los requisitos

## Formato de las historias de usuarios.

**Fig.** 7 ejemplo de historia de usuario



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/history-user-100x79.PNG VISITA:24-sep-2018)

* El Product owner tiene la visión del producto a desarrollar y descompone la

funcionalidad del sistema en un conjunto de características que se recogen en forma de lista priorizada en el Product Backlog (grooming)

* En la planificación de cada Sprint se seleccionan las características a implementar (Sprint backlog) en el Sprint (Sprint execution), se estima tiempo y esfuerzo y se adquiere el compromiso de realizarlo
* Durante la ejecución del Sprint, se llevan a cabo las tareas necesarias para implementar las características del Sprint backlog y se mantienen reuniones breves diarias para revisar, obtener feedback y planificar (Daily scrum)
* Al finalizar cada ejecución de Sprint, el equipo produce una mejora o incremento del producto que es potencialmente entregable al cliente.
* El equipo y los stakeholders revisan el producto entregado (Sprint review) y recoge el feedback.
* El equipo revisa el proceso Scrum que están utilizando y se adapta si es necesario (Scrum retrospective)
* El ciclo se repite con la selección del nuevo conjunto de características a implementar en un nuevo Sprint
* Tras una serie de Sprints, el equipo completa la implementación del producto.

# 3.9 HERRAMIENTA DE MODELAMIENTO DE LAS FASES DE ANALISIS Y DISEÑO DEL PROYECTO: Lenguaje Unificado de Modelado – UML

# El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

“Sirve para brindar a arquitectos de sistemas, ingenieros y desarrolladores de software las herramientas para el análisis, el diseño y la implementación de sistemas basados en software, así como para el modelado de procesos de negocios y similares.”[[4]](#footnote-4)

## 3.9.1 Diagrama de Caso de Uso.

“Los diagramas de caso de uso son uno de los cinco tipos de diagramas en UML para modelar aspectos dinámicos de sistemas (diagramas de actividad, diagramas de estados, diagramas de secuencia y diagramas de colaboración son otros cuatro tipos de diagramas en UML para modelar los aspectos dinámicos de un sistema). Los diagramas de casos de uso son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. ” [[5]](#footnote-5)

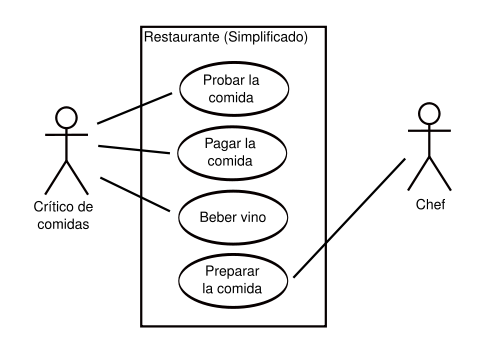
Se aplican los diagramas de casos de uso para modelar las vistas de casos de uso de un sistema. Para la mayor parte, esto involucra el modelado el contexto de un sistema, subsistema, o clase, o modelar las necesidades del comportamiento de esos elementos.

Los diagramas de casos de uso son importantes para visualizar, especificar, y documentar el comportamiento de un elemento. Ellos hacen sistemas, subsistemas, y clases entendibles para presentar una vista exterior de cómo estos elementos pueden ser usados dentro del contexto.

### 3.9.1.1 Simbolos de Caso de Uso.

**3.9.1.1.1** Sistemas:El rectángulo representa los límites del sistema que contiene los casos de uso. Los actores se ubican fuera de los límites del Sistema.

**Fig**. 8 símbolos de caso de uso

**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/3.png)**

( **Ref** .<https://upload.wikimedia.org/wikipedia/commons/thumb/8/80/UML_diagrama_caso_de_uso.svg/747px-UML_diagrama_caso_de_uso.svg.png> VISITA: 08/sep./2018)

3.9.1.1.2 Caso de Uso: Se representan con óvalos. La etiqueta en el óvalo indica la función del sistema.

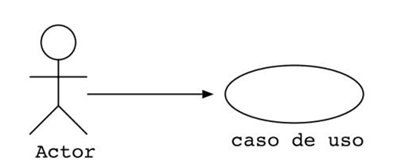
**Fig**. 9 Casos de Uso

**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/4.png)**

**(Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/5.png> VISITA: 08-sep-2018)

#### ***3.9.1.1.3***Actor: Un diagrama de caso de uso contiene los símbolos del actor y del caso de uso, junto con líneas conectoras. Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.

**Fig**. 10 Diagrama que contiene al actor y el caso de uso

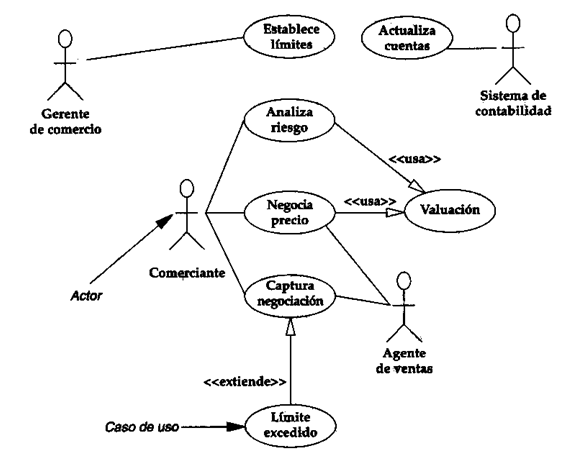
**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/5.png)**

(**Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/5.png> VISITA:08-sep-2018)

#### ***3.9.1.1.4***Relaciones

Las relaciones entre un actor y un caso de uso, se dibujan con una línea simple. Para relaciones entre casos de uso, se utilizan flechas etiquetadas “incluir” o “extender.” Una relación “incluir” indica que un caso de uso es necesitado por otro para poder cumplir una tarea. Una relación “extender” indica opciones alternativas para un cierto caso de uso.

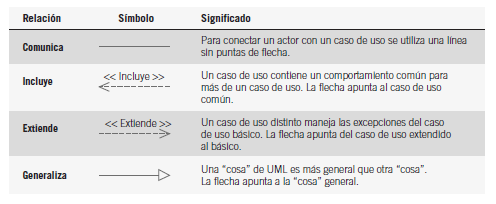
**Fig**. 11 Ejemplo de Casos de Uso

**[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/esta.png)**(**Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/esta.png?w=646> VISITA:08-sep-2018)

### 3.9.2.2 Relaciones de Caso de Uso

Las relaciones activas se conocen como relaciones de comportamiento y se utilizan principalmente en los diagramas de casos de uso. Hay cuatro tipos básicos de relaciones de comportamiento: comunica, incluye, extiende y generaliza.

**Fig**. 12 Relaciones entre los Casos de Uso

[](https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/8.png)

(**Ref**.<https://ingsotfwarekarlacevallos.files.wordpress.com/2015/07/8.png> VISITA:08-sep-2018)

## 3.10.3 DIAGRAMA DE CLASES

“[En ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software), un diagrama dees un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.”[[6]](#footnote-6)

Los diagramas de clases ofrecen una serie de beneficios para toda organización. Usa los diagramas de clases UML para:

* Ilustrar modelos de datos para sistemas de información, sin importar qué tan simples o complejos sean.
* Comprender mejor la visión general de los esquemas de una aplicación.
* Expresar visualmente cualesquier necesidades específicas de un sistema y divulgar esa información en toda la empresa.
* Crear diagramas detallados que resalten cualquier código específico que será necesario programar e implementar en la estructura descrita.
* Ofrecer una descripción independiente de la implementación sobre los tipos empleados en un sistema que son posteriormente transferidos entre sus componentes.

### 3.10.3.1 Componentes Basicos de Diagrama de Clases

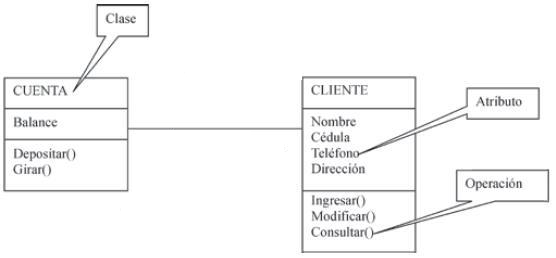
El diagrama de clases estándar está compuesto por tres partes:

**3.10.3.1.1 Sección superior:** Contiene el nombre de la **clase**. Esta sección siempre es necesaria, ya sea que estés hablando del clasificador o de un objeto.

**3.10.3.1.2** **Sección central:** Contiene los **atributos** de la clase. Usa esta sección para describir cualidades de la clase. Esto solo es necesario al describir una instancia específica de una clase.

**3.10.3.1.3** **Sección inferior:** Incluye **operaciones** de clases (métodos). Esto está organizado en un formato de lista. Cada operación requiere su propia línea. Las operaciones describen cómo una clase puede interactuar con los datos.

**Fig**. 13 ejemplo de clases



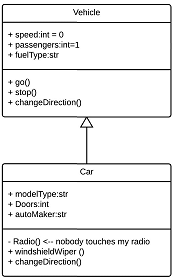
(**Ref**. Elaboración Propia)

### 3.10.3.2 Interacciones

El término "interacciones" se refiere a múltiples relaciones y enlaces que pueden existir en diagramas de objetos y de clases. Algunas de las interacciones más comunes incluyen:

3.10.3.2.1 HERENCIA: El proceso en el que una subclase o clase derivada recibe la funcionalidad de una superclase o clase principal, también se conoce como "generalización". Se simboliza mediante una línea de conexión recta con una punta de flecha cerrada que señala a la superclase.

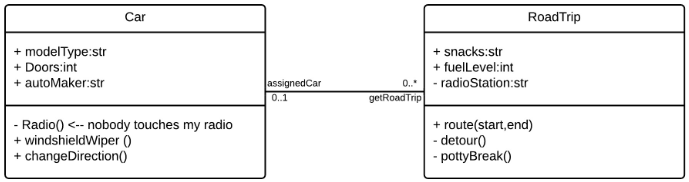
**Fig**. 14 Ejemplo de herencia



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/uml/class-diagram/class-diagram-inheritance-175x279.PNG VISITA:08-sep-2018)

* + - * 1. ASOCIACIÓN BIDIRICCIONAL La relación predeterminada entre dos clases. Ambas clases están conscientes una de la otra y de la relación que tienen entre sí. Esta asociación se representa mediante una línea recta entre dos clases.

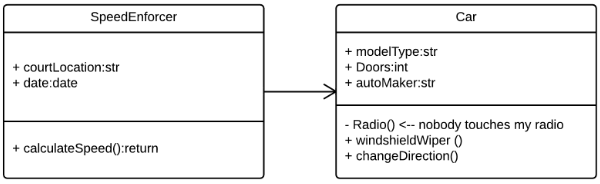
**Fig**. 15 Ejemplo de asociación bidireccional



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/uml/class-diagram/class-diagram-bi-directional-association-689x182.PNG VISITA: 08-sep-2018)

* + - * 1. ASOCIACIÓN UNIDIRECCIONAL: Una relación un poco menos común entre dos clases. Una clase está consciente de la otra e interactúa con ella. La asociación unidireccional se dibuja con una línea de conexión recta que señala una punta de flecha abierta desde la clase "knowing" a la clase "known".

**Fig**. 16 Ejemplo Asociación unidireccional



(**Ref**.https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/uml/class-diagram/class-diagram-unidirectional-association-600x184.PNG VISITA:08-sep-2018)

## 3.10.4 DIAGRAMA DE COMPONENTES

El diagrama de componentes proporciona una visión física de la construcción del sistema de información. Muestra la organización de los componentes software, sus interfaces y las dependencias entre ellos.

### 3.10.4.1 descripci**ón**

Como ya se ha indicado, los elementos de estos diagramas son los componentes software y las dependencias entre ellos.

“Un componente es un módulo de software que puede ser código fuente, código binario, un ejecutable, o una librería con una interfaz definida. Una interfaz establece las operaciones externas de un componente, las cuales determinan una parte del comportamiento del mismo. Además se representan las dependencias entre componentes o entre un componente y la interfaz de otro, es decir uno de ellos usa los servicios o facilidades del otro.”[[7]](#footnote-7)

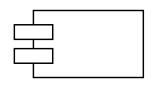
Estos diagramas pueden incluir paquetes que permiten organizar la construcción del sistema de información en subsistemas y que recogen aspectos prácticos relacionados con la secuencia de compilación entre componentes, la agrupación de elementos en librerías, etc.

### 3.10.4.2 Notaci**ón**

### 3.10.4.3 Componentes

Un componente se representa como un rectángulo, con dos pequeños rectángulos superpuestos perpendicularmente en el lado izquierdo.

**Fig**. 17 Componente rectangular



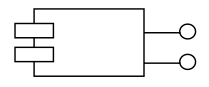
(**Ref**.http://manuel.cillero.es/wp-content/uploads/2013/11/componente.png?x58486 VISITA:08-sep-2018)

Para distinguir distintos tipos de componentes se les puede asignar un estereotipo, cuyo nombre estará dentro del símbolo: << ... >>

### 3.10.4.4 Interfaz

Se representa como un pequeño círculo situado junto al componente que lo implementa y unido a él por una línea continua. La interfaz puede tener un nombre que se escribe junto al círculo. Un componente puede proporcionar más de una interfaz.

**Fig**. 18 Interfaz representada con un pequeño circulo



(**Ref**. http://manuel.cillero.es/wp-content/uploads/2013/11/interfaz.png?x58486 (08-sep-2018)

**3.10.4.5** **Paquete**

Un paquete se representa con un icono de carpeta.

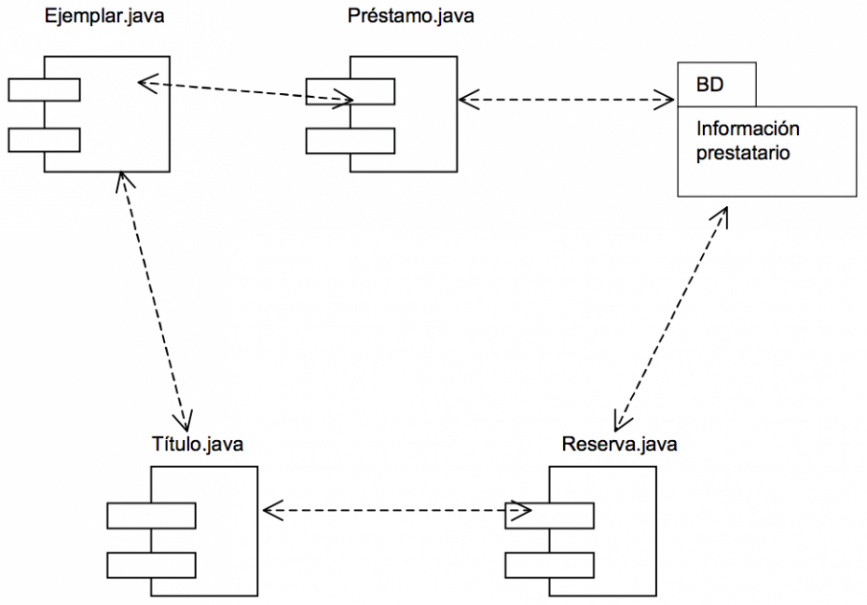
**3.10.4.6 Relacion de Dependencia**

Una relación de dependencia se representa mediante una línea discontinua con una flecha que apunta al componente o interfaz que provee del servicio o facilidad al otro. La relación puede tener un estereotipo que se coloca junto a la línea, entre el símbolo: <<...>>

### Ejemplo

Sistema encargado de la gestión de los préstamos y reservas de libros y revistas en una biblioteca. El lenguaje de desarrollo será Java, y los accesos a la información del prestatario serán mediante un paquete de Base de Datos.

**Fig**. 19 Esta notación es la más habitual, pero MÉTRICA Versión 3 no exige su utilización

[](http://manuel.cillero.es/wp-content/uploads/2013/11/ejemplo-diagrama-de-componentes.png?x58486)

(**Ref**.http://manuel.cillero.es/wp-content/uploads/2013/11/ejemplo-diagrama-de-componentes.png?x58486 VISITA:08-sep-2018)

## 3.10.5 DIAGRAMA DE DESPLIEGUE

“El diagrama de despliegue permite mostrar la arquitectura en tiempo de ejecución del sistema respecto al hardware y software. Este se utiliza en el diseño y la implementación y se pueden distinguir componentes y nodos, así como las relaciones entre estos.” [[8]](#footnote-8)

* La arquitectura lógica del diagrama de despliegue es la siguiente:
* Capa logica de presentacion y de aplicaciones en la computadora del cliente, en su almacenamiento o en su servicio.
* La presentacion en la computadora del cliente, la logica de aplicaciones en un servidor de la aplicación y el almacenamiento en un servidor de los subsistemas se ira realizando de manera distribuida cada vez mas.

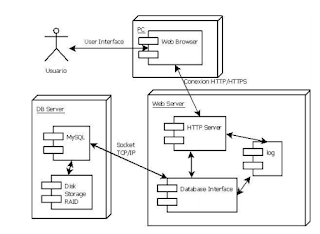
3.10.5.1 Características

* Utilizando en las etapas del diseño de proceso.
* Se refinan a lo largo del proceso de desarrollo.
* Utiles en el proceso de instalacion del sistema.

Este solo presenta la estructura del sistema solo en tiempos de ejecucion pero no en tiempo de desarrollo o compilacion. Se utiliza mucho para la aplicación distribuidas u online.

* Modela aspectos fisicos de un sistema.
* Modela la vista de despliegue estatica de un sistema.
* Modela una configuracion de nodos y los componentes que residen en ellos.
* Modela la topologia del hardware donde se ejecuta el sistema.

**Fig**. 20 Ejemplo diagrama de despliegue



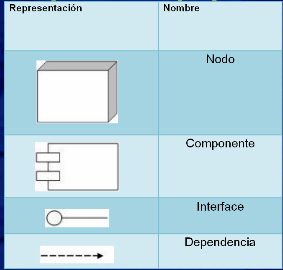
(**Ref**.http://3.bp.blogspot.com/-pIskqASgsbw/VbXaR3ixboI/AAAAAAAAA9I/fVwToVHpVQY/s320/CAS.png  VISITA: 08-sep-2018).

**3.10.5.2** Los Elementos que lo Componen son:

* Relaciones de dependencia, generalizando,asociaacion y realizacion.
* Pueden contener los componentes que residen en los nodos.

NODOS**:** Representan objetos físicos existentes en tiempo de ejecución, sirven para modelar recursos que tiene la memoria y capacidad de proceso. Los nodos se representan mediante paralelepípedos rectangulares.   
  
**3.10.5.3** Relaciones de Dependencia: Entre los nodos se establecen relaciones que significan que existen comunicación entre estos. Se representan mediante líneas continuas y se puede hacer con un estereotipo que indica el tipo de comunicación.  
  
**3.10.5.4** Componente: Se puede ejecutar si se utilizan los recursos de un nodo que pueden estar contenidos en este.

**Fig**. 21 Elementos que lo componen



(**Ref**.http://3.bp.blogspot.com/-bLyqMRurVaw/VbXZ4E8gbfI/AAAAAAAAA9A/MzP-7YHUzIU/s1600/M.png VISITA: 08-sep-2018)

# 3.10.6 HERRAMIENTAS DE DESARROLLO

**3.10.6.1** Php**:** “Acrónimo recursivo en inglés de PHP Hypertext Preprocessor(preprocesador de hipertexto), es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [de propósito general](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general) de [código del lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) originalmente diseñado para el [desarrollo web](https://es.wikipedia.org/wiki/Desarrollo_web) de [contenido dinámico](https://es.wikipedia.org/wiki/Contenido_din%C3%A1mico). Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento [HTML](https://es.wikipedia.org/wiki/HTML) en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.”[[9]](#footnote-9)

PHP ha evolucionado por lo que ahora incluye también una interfaz de [línea de comandos](https://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos) que puede ser usada en [aplicaciones gráficas](https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario) independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

**3.10.6.2** Servidor Web Apache**.** -El servidor Apache HTTP, también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

“Su misión es crítica, ya que es el encargado de aceptar las peticiones de páginas (o recursos en general) que provienen de los visitantes que acceden a nuestro sitio web y gestionar su entrega o denegación, de acuerdo a las políticas de seguridad establecidas. Esto, que puede parecer simple, implica muchas facetas y funcionalidades que debe cubrir, como pueden ser: Atender de manera eficiente, ya que puede recibir un gran número de peticiones HTTP, restricciones de acceso a los ficheros que no se quieran ‘exponer’, gestión de autentificaciones de usuarios o filtrado de peticiones según el origen de éstas, manejar los errores por páginas no encontradas, informando al visitante y/o redirigiendo a páginas predeterminadas o gestión de la información a transmitir en función de su formato e informar adecuadamente al navegador que está solicitando dicho recurso.”[[10]](#footnote-10)

**3.10.6.3** XAMPP:  es un servidor independiente de plataforma de código libre. Permite instalar de forma sencilla Apache propio ordenador, sin sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD o FileZilla FTP Serve, etc. entre muchas cosas más.

“Si alguna vez has intentado instalar Apache, sabes que no es una tarea fácil, sin embargo, con XAMPP todo es diferente. Una de las ventajas de usar XAMPP es que su instalación es de lo más sencilla, basta descargarlo, extraerlo y comenzar a usarlo.”[[11]](#footnote-11)

En si es una herramienta de desarrollo trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener que Accesar a internet.

# 3.10.7 HERRAMIENTAS DE DISEÑO Y DESARROLLO

3.10.7.1 Visio 2013: “Es un software de dibujo vectorial, las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más, que permiten iniciar al usuario en los lenguajes de programación.” **[[12]](#footnote-12)** Nos permitió realizar los diagramas de UML para demostrar el análisis y diseño del sistema.

3.10.7.2 Sqlyog: “Es una excelente interfaz gráfica diseñada especialmente para trabajar de forma más rápida y cómoda con el servidor de base de datos [MySQL](https://www.ecured.cu/MySQL). El programa está orientado a usuarios que ya tienen ciertos conocimientos de [SQL](https://www.ecured.cu/SQL) y necesitan un intérprete gráfico sin excesivas florituras, ágil y funcional. SQLyog te permite administrar usuarios y permisos, y realizar múltiples peticiones a base de datos.” [[13]](#footnote-13) También puedes insertar fácilmente, mediante plantillas, peticiones que realices con frecuencia, como por ejemplo creación de tablas. Otras opciones son la exportación de datos en formato [CSV](https://www.ecured.cu/CSV), [HTML](https://www.ecured.cu/HTML) y [XML](https://www.ecured.cu/XML), la optimización de bases de datos, la posibilidad de guardar [scripts](https://www.ecured.cu/Scripts) de [SQL](https://www.ecured.cu/SQL) como favoritos. Soporte Unicode/UTF8 completo, Productividad desarrollador / usuario, Documentación de esquema [HTML](https://www.ecured.cu/HTML), Atajos para generar sentencias [SQL](https://www.ecured.cu/SQL) [DML](https://www.ecured.cu/index.php?title=DML&action=edit&redlink=1) desde definición de esquema, Editor de consultas con pestañas múltiples y editor de resultados, Ejecución múltiple de consultas, Ejecución de consultas multihilo - Posibilidad de detener consultas grandes, Plantillas [SQL](https://www.ecured.cu/SQL), Interfaz de cuadrícula estilo [Excel](https://www.ecured.cu/Excel) para ver/actualizar los resultados, Editor de Blob multiformato, Se pueden ver datos en modo cuadrícula o texto, Exportación de Resultado/Datos a [CSV](https://www.ecured.cu/CSV) / [XML](https://www.ecured.cu/XML) / [HTML](https://www.ecured.cu/HTML) / [Excel](https://www.ecured.cu/Excel).

## 3.10.7.3 MYSQL Workbench: Es un software creado por [la empresa](https://www.monografias.com/trabajos11/empre/empre.shtml) Sun Microsystems, esta herramienta permite modelar diagramas de Entidad-Relación para bases de datos MySQL.

“Con esta herramienta se puede elaborar una representación visual de las tablas, vistas, [procedimientos](https://www.monografias.com/trabajos13/mapro/mapro.shtml) almacenados y claves foráneas de la base de datos. Además, es capaz de sincronizar el [modelo](https://www.monografias.com/trabajos/adolmodin/adolmodin.shtml) en [desarrollo](https://www.monografias.com/trabajos12/desorgan/desorgan.shtml) con la base de datos real. Se puede realizar una [ingeniería](https://www.monografias.com/trabajos14/historiaingenieria/historiaingenieria.shtml) directa e ingeniería inversa para exportare e importar el esquema de una base de datos ya existente el cual haya sido guardado o hecho copia de [seguridad](https://www.monografias.com/trabajos/seguinfo/seguinfo.shtml) con MySQL Administrador.” [[14]](#footnote-14)

MySQL Workbench puede generar también el guion necesario para crear la base de datos que se ha dibujado en el esquema; es compatible con los [modelos](https://www.monografias.com/trabajos/adolmodin/adolmodin.shtml) de base de datos de DBDesigner 4 y soporta las novedades incorporadas en MySQL 5.x

## 3.10.7.4 Sublime Text:

Es un editor de código multiplataforma, ligero y con pocas concesiones a las florituras. Es una herramienta concebida para programar sin distracciones. Su interfaz de color oscuro y la riqueza de coloreado de la sintaxis, centra nuestra atención completamente.

[Sublime Text](http://www.sublimetext.com/) permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla.

Soporta un gran número de lenguajes (C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML).

“Este programa dispone de auto-guardado, muchas opciones de personalización especializada, cuenta con un buen número de herramientas para la edición del código y automatización de tareas. Soporta macros, Snippets y auto completar, entre otras funcionalidades. Algunas de sus características son ampliables mediante plugins.” [[15]](#footnote-15)

## 3.10.7.5 **Adobe Photoshop CS6;**

“Adobe Photoshop es un [editor de gráficos rasterizados](https://es.wikipedia.org/wiki/Editor_de_gr%C3%A1ficos_rasterizados) desarrollado por [Adobe Systems Incorpórate](https://es.wikipedia.org/wiki/Adobe_Systems_Incorporated). Usado principalmente para el [retoque](https://es.wikipedia.org/wiki/Retoque_fotogr%C3%A1fico) de [fotografías](https://es.wikipedia.org/wiki/Fotograf%C3%ADa) y gráficos, su nombre en español significa literalmente ‘taller de fotos’. Es líder mundial del mercado de las [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) de edición de imágenes y domina este sector de tal manera que su nombre es ampliamente empleado como sinónimo para la edición de imágenes en general.“[[16]](#footnote-16)

Muchas empresas (en especial empresas web), utilizan en este programa para realizar los logotipos con los que se identificarán.

Entre las varias funciones con que cuenta este programa, se encuentran, aquellas que permiten realizar “marcas de agua” o firmas, en ciertas imágenes, mismas que son utilizadas por varias empresas para que sus imágenes, no sean utilizadas sin autorización por otros usuarios cuando son puestas en la web.

Este programa permite guardar los archivos además del formato nativo del programa, en los principales formatos de imágenes. Es el caso de los formatos como: PostScript, EPS, DCS, BMP, GIF, JPEG, PICT, PIFF, PNG, PDF, IFF, PCX, RAW, TGA, Scitex CT, Filmstrip, FlashPix, entre otros.**[[17]](#footnote-17)**

## 3.10.7.6 **Navegador Google Chrome;**

Google Chrome es un [navegador web](http://www.alegsa.com.ar/Dic/navegador.php) de [código abierto](http://www.alegsa.com.ar/Dic/codigo%20abierto.php) desarrollado por [Google](http://www.alegsa.com.ar/Dic/google.php).  
“Este navegador está constituido por componentes de otros navegadores de código abierto como WebKit y Mozilla. El objetivo de Google Chrome es lograr un navegador con una interfaz sencilla, y una muy buena estabilidad, velocidad y seguridad.”**[[18]](#footnote-18)**

## 3.10.7.7 **Phpstorm Ide. -**

IntelliJ IDEA es un [entorno de desarrollo integrado](https://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado)(IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: edición para la comunidad[1](https://es.wikipedia.org/wiki/IntelliJ_IDEA#cite_note-1)​ y edición comercial. IntelliJ IDEA no está basada en [Eclipse](https://es.wikipedia.org/wiki/Eclipse_(software)) como MyEclipse u Oracle Enterprise Pack para Eclipse.**[[19]](#footnote-19)**

## 3.10.7.8 **Dia. -**

“Dia es una [aplicación informática](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) de propósito general para la creación de diagramas, creada originalmente por Alexander Larsson[1](https://es.wikipedia.org/wiki/Dia_(programa)#cite_note-1)​, y desarrollada como parte del proyecto [GNOME](https://es.wikipedia.org/wiki/GNOME) .” [[20]](#footnote-20) Está concebido de forma modular, con diferentes paquetes de formas para diferentes necesidades.

# CAPITULO 4

# FASE DE ANALISIS Y DISEÑO

# INTRODUCCIÓN:

El análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorar con métodos y procedimientos más adecuados. El desarrollo de sistemas tiene dos componentes.

Es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema.

En este capítulo se muestra todo el proceso de la fase de análisis y diseño mediante la recolección de información previa utilizando las técnicas de recolección de información tales como:

* Encuestas
* Entrevistas

Se utilizo la Metodología Scrum y los diferentes diagramas DE UML que están a continuación.

* Historias de usuario.
* Descripción de actores.
* Diagrama casos uso
* Diagrama de clases
* Diagrama entidad relación

# 4.1 HISTORIAS DE USUARIO DEL PROYECTO

Son descripciones rotas y esquemáticas, que resumen la necesidad concreta de un usuario al utilizar un producto o servicio como también a la solución que lo satisface.

Su función principal es identificar problemas percibidos y proponer soluciones y de esa manera estimar los esfuerzos necesarios para las ideas propuestas.

Mediante el cuestionario realizado a las personas involucradas se sacarán historias de usuario concretas que nos ayudarán a determinar los pasos a seguir en el desarrollo del sistema.

|  |  |
| --- | --- |
| **HISTORIA DE USUARIO** | **CODIGO: 1** |
|
| **ROL DE USUARIO: E**studiante | **FECHA:** |
| **DESCRIPCIÓN: Y**o como usuario(estudiante) , me gustaría verlos requisitos necesarios de la pasantía, de modo que con esos requisitos poder inscribirme a la pasantía. | |
|
|
| **OBSERVACIÓN: N**inguna | |
|
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 2** |
|
| **ROL DE USUARIO:  E**studiante | **FECHA:** |
| **DESCRIPCIÓN: Y**o como (estudiante)necesito saber un lugar para realizar mi pasantía, de modo que si hay convenios con alguna empresa. | |
|
|
| **OBSERVACIÓN: N**inguna | |
|
|  |  |
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO:3** |
|
| **ROL DE USUARIO:  T**utor de pasantía | **FECHA:** |
| **DESCRIPCIÓN:** **Y**o como (tutor) necesito saber el tiempo de entrega de la documentación de la pasantía de modo que no tenga contratiempos con la presentación. | |
|
|
| **OBSERVACIÓN: N**inguna | |
|
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 4** |
|
| **ROL DE USUARIO:  T**utor de pasantia | **FECHA:** |
| **DESCRIPCIÓN: Y**o como tutor necesito llevar un control de la pasantia del estudiante. | |
|
|
| **OBSERVACIÓN: N**inguna | |
|
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 5** |
|
| **ROL DE USUARIO:  E**studiante | **FECHA:** |
| **DESCRIPCIÓN: Y**o como (estudiante) necesito saber el tiempo mínimo(horas) que debo estar en mi pasantia. De modo que pueda controlar mis horas. | |
|
|
| **OBSERVACIÓN: N**inguna | |
|
|  |  |
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 6** |
|
| **ROL DE USUARIO:  E**studiante | **FECHA:** |
| **DESCRIPCIÓN: Y**o como (estudiante) necesito saber a quien informarle sobre mis quejas en cuanto a mi pasantia, de modo así solucionarlas. | |
|
|
| **OBSERVACIÓN: N**inguna | |
|
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 7** |
|
| **ROL DE USUARIO:  E**ncargado de registros | **FECHA:** |
| **DESCRIPCIÓN:** Yo como (encargado de registros) me gustaría que el estudiante venga con todos los requisitos para la toma de pasantia, de modo que acortaría el tiempo. | |
|
|
| **OBSERVACIÓN:** | |
|
|  |  |
| **HISTORIA DE USUARIO** | **CODIGO: 8** |
|
| **ROL DE USUARIO:  E**studiante | **FECHA:** |
| **DESCRIPCIÓN:** **Y**o como (estudiante) me gustaría saber quien es mi tutor de pasantia para pedirle información, de modo que esto facilitaría todo. | |
|
|
| **OBSERVACIÓN:** | |
|

# 4.2 MODELO DE CASOS DE USO

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto, los casos de uso determinan los requisitos funcionales del sistema basados en las historias de usuarios descritos nteriormente.

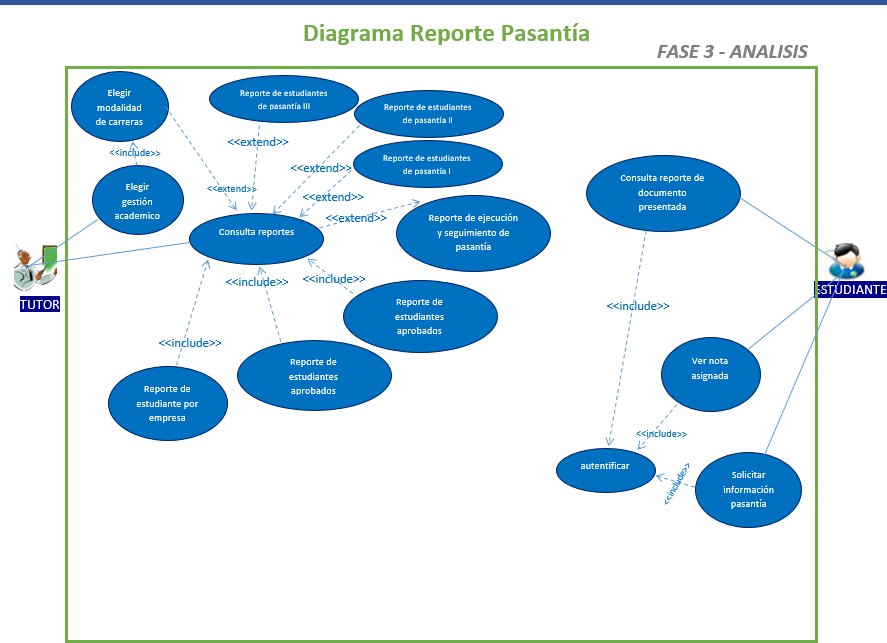
## 4.2.1 IDENTIFICACIÓN DE ACTORES

Una vez obtenido los requerimientos del sistema también se identifica a los actores que interactuan con él mismo, donde la identificación de los actores del modelo de casos de uso permite.

Los actores del sistema son los siguientes:

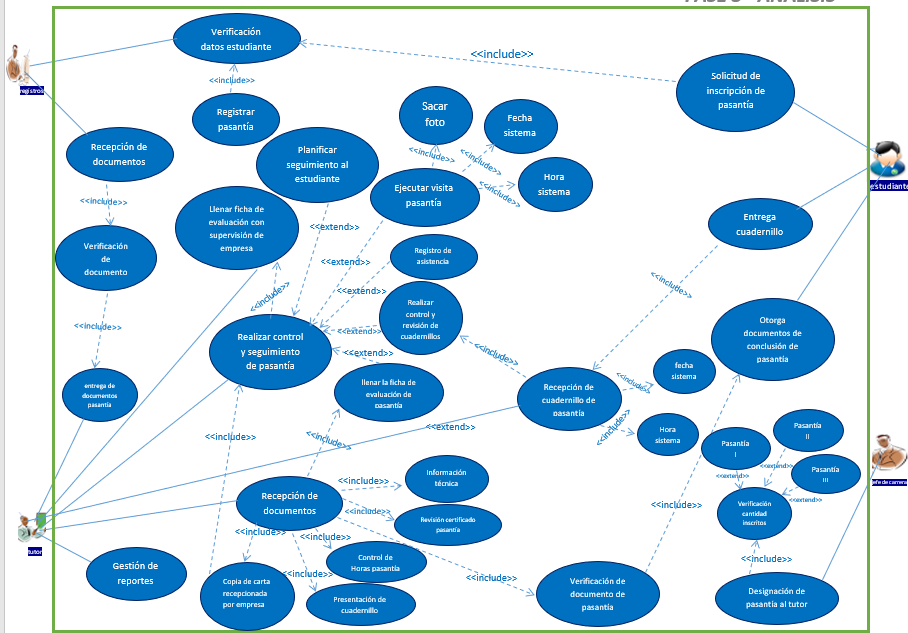
* + - 1. **Registro,** es la persona encargada del manejo sistema y es el que verifica la documentacion y la recepciona.
      2. **Tutor,** es el individuo que es el encargado de controlar la pasantia. Asimismo, es el que registra las notas del pasante en el sistema.
      3. **Estudiante,** son los que tendran el acceso de ver requisitos de pasantia y en que estado se enuentra.
      4. **Jefe de Carrera,** es el individuo que es el encargado de controlar al tutor sobre la pasantia.

**Fig**. 22 Descripción de actividades de los actores Tutor y Estudiante



(**Ref**. Elaboración Propia)

**Fig**. 23 Descripción de actividades de los actores jefe de Carrera y Registros



***DIAGRAMA GENERAL DE PASANTIA***

(**Ref**. Elaboración Propia)

## 4.2.2 DESCRIPCIÓN CASO DE USO: A continuación, se describirán cada uno de los escenarios más importantes del sistema Internship.

### 4.3.2 .1 Escenario 1: Reportes de Pasantías

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Reporte de Pasantías** | 1 |
| Actores | 1. Tutor | |
| **Tipo** | **Secundario** | |
| Referencias | Tutor ingresa al sistema con su usuario e imprime información detallada sobre el pasante. | |
| **Propósito** | | |
| Generar reportes completos inherentes a la pasantía del estudiante. | | |
| **Descripción** | | |
| 1º El tutor ingresa el carnet de identidad del estudiante en el sistema.  2º El sistema verifica los datos del estudiante y solicita el tipo de reporte que necesita.  3º El tutor elige el tipo de reporte:   * Reporte de datos personales estudiante * Reporte de datos de la empresa * Reporte documentación presentada por el estudiante * Documentación faltante del estudiante * Reporte de notas estudiante   4º El sistema genera los informes requeridos en el sistema.  5º El sistema imprime el reporte para el solicitante. | | |

(**Ref**. Elaboración Propia)

### 4.3.2.2 Escenario 2: Reporte de Estudiante

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Reporte de Pasantías** | 1 |
| Actores | 1. Estudiante | |
| **Tipo** | **Secundario** | |
| Referencias | Estudiante ingresa al sistema con su usuario e imprime información detallada sobre la información que tiene permitida. | |
| **Propósito** | | |
| Generar reportes específicos de la pasantía del estudiante. | | |
| **Descripción** | | |
| 1º El estudiante ingresa al sistema con su número de carnet.  2º El sistema verifica los datos del estudiante y solicita el tipo de reporte que necesita.  El estudiante elige el tipo de reporte:   * Reporte de notas estudiante * Reporte de documentación presentada * Documentación faltante del estudiante   3º Genera los informes requeridos en el sistema.  4º Imprime el reporte para el solicitante. | | |

(**Ref**. Elaboración Propia)

### 4.3.2.3 Escenario 3: Reporte de Jefe de Carrera

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Reporte de Pasantías** | 1 |
| Actores | 1. Jefe de carrera | |
| **Tipo** | **Secundario** | |
| Referencias | Jefe de Carrera debe revisar los reportes de los tutores. | |
| **Propósito** | | |
| Generar reportes específicos inherentes a la pasantía del estudiante. | | |
| **Descripción** | | |
| 1º El Jefe de Carrera ingresa el carnet de identidad del estudiante en el sistema.  2º EL sistema verifica los datos del estudiante y solicita el tipo de reporte que necesita.  3º El Jefe de Carrera elige el tipo de reporte:   * Reporte de datos personales estudiante * Reporte de datos de la empresa * Reporte documentación presentada por el estudiante * Documentación faltante del estudiante * Reporte de notas estudiante   4º El sistema genera los informes requeridos en el sistema.  5º El sistema imprime el reporte para el solicitante. | | |

(**Ref**. Elaboración Propia)

### 4.3.2.4 Escenario 4: Reporte de Registro

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Reporte de Pasantías** | 1 |
| Actores | 1. Registros | |
| **Tipo** | **Secundario** | |
| Referencias | Registros debe tener la información de los pasantes en archivos | |
| **Propósito** | | |
| Generar reportes específicos DE la pasantía del estudiante. | | |
| **Descripción** | | |
| 1º El personal de registros ingresa el carnet de identidad del estudiante en el sistema.  2º El sistema verifica los datos del estudiante y solicita el tipo de reporte que necesita.  3º El Registro elige el tipo de reporte:   * Reporte de datos personales estudiante * Reporte de datos de la empresa * Reporte documentación presentada por el estudiante * Documentación faltante del estudiante * Reporte de notas estudiante   4º El sistema genera los informes requeridos en el sistema.  5º El sistema imprime el reporte para el solicitante. | | |

(**Ref**. Elaboración Propia)

### 4.3.2.6 Escenario 6: Revisión documentos Área de Registro

|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Diagrama revisión documentos área de registros** | 1 |
| Actores | 1. Registros 2. Tutor | |
| Tipo | Primario | |
| Propósito | | |
| Revisar y cotejar documentación presentada por tutor. | | |
| Descripción | | |
| 1º Solicita generación de reportes de documentos presentados por el estudiante  2º El sistema genera reporte  verifican los datos del estudiante y documentación presentada  3º Guardas cambios. | | |

### 4.3.2.7 Escenario 7: Seguimiento Tutor al Pasante

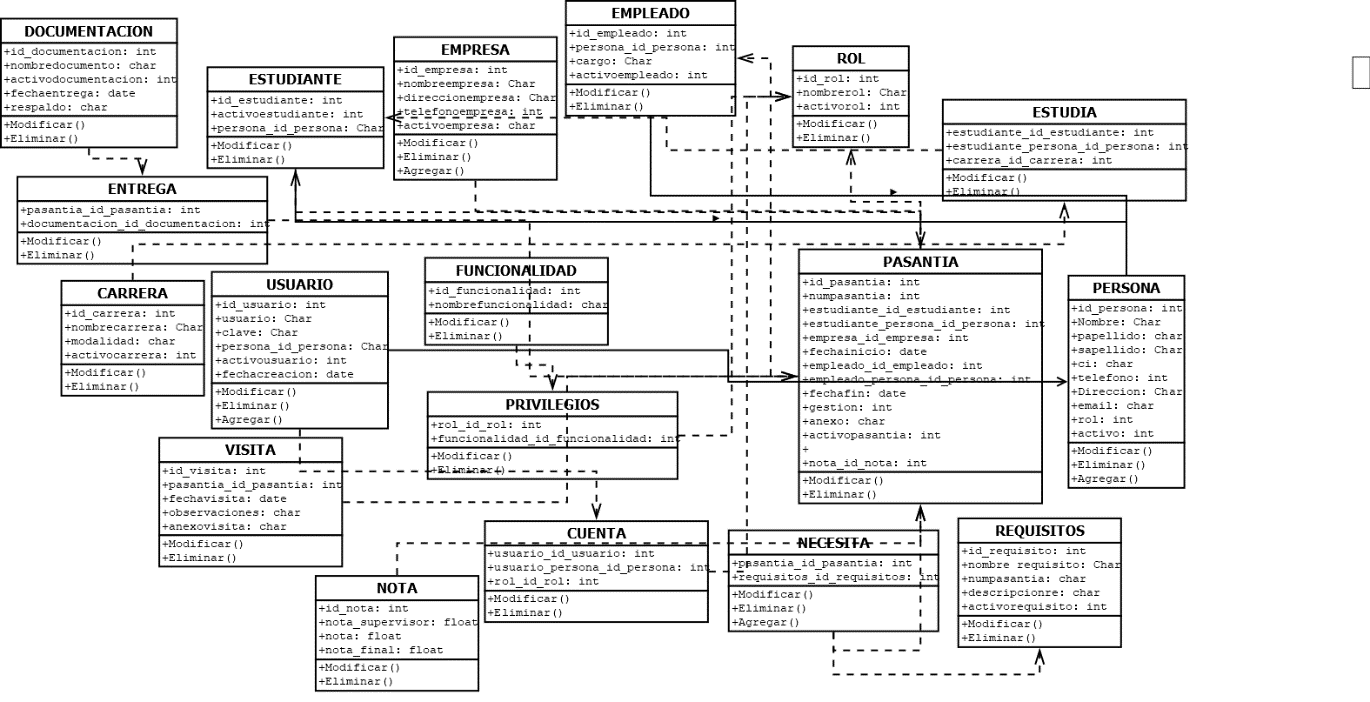
|  |  |  |
| --- | --- | --- |
| **Caso de Uso** | **Diagrama seguimiento tutor al pasante** | 1 |
| Actores | 1. Tutor 2. Registro | |
| Tipo | Primario | |
| Propósito | | |
| Seguimiento de pasantía | | |
| Descripción | | |
| 1º Tutor recepciona el cuadernillo de pasante.  2º Procede a calificar el desempeño del pasante en la semana de acuerdo a la presentación de cuadernillos.  3º Registra la fecha de revisión de cuaderno en sistema, las observaciones.  4º Sistema guarda cambios efectuados por el tutor.  5º El tutor, realiza visitas sorpresa a la empresa del pasante.  6º Sistema captura foto y guarda con fecha y hora de visita.  7º Especificar que registrara el sistema: una foto ; una ficha.  8º Tutor genera ficha evaluación del estudiante.  9º Llena la ficha de evaluación con supervisión de la empresa. | | |

(**Ref**. Elaboración Propia)

# 4.3 DIAGRAMA **DE CLASES**

Un diagrama de Clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Nos sirve para visualizar las relaciones entre las clases que involucran el sistema, también nos permitirá representar gráficamente y de manera estática la estructura general de un sistema, mostrando cada una de las clases y sus interacciones (como herencias, asociaciones, etc.)~~.~~

## 4.3.2 ULTIMA VERSIÓN DEL DIAGRAMA DE CLASES:

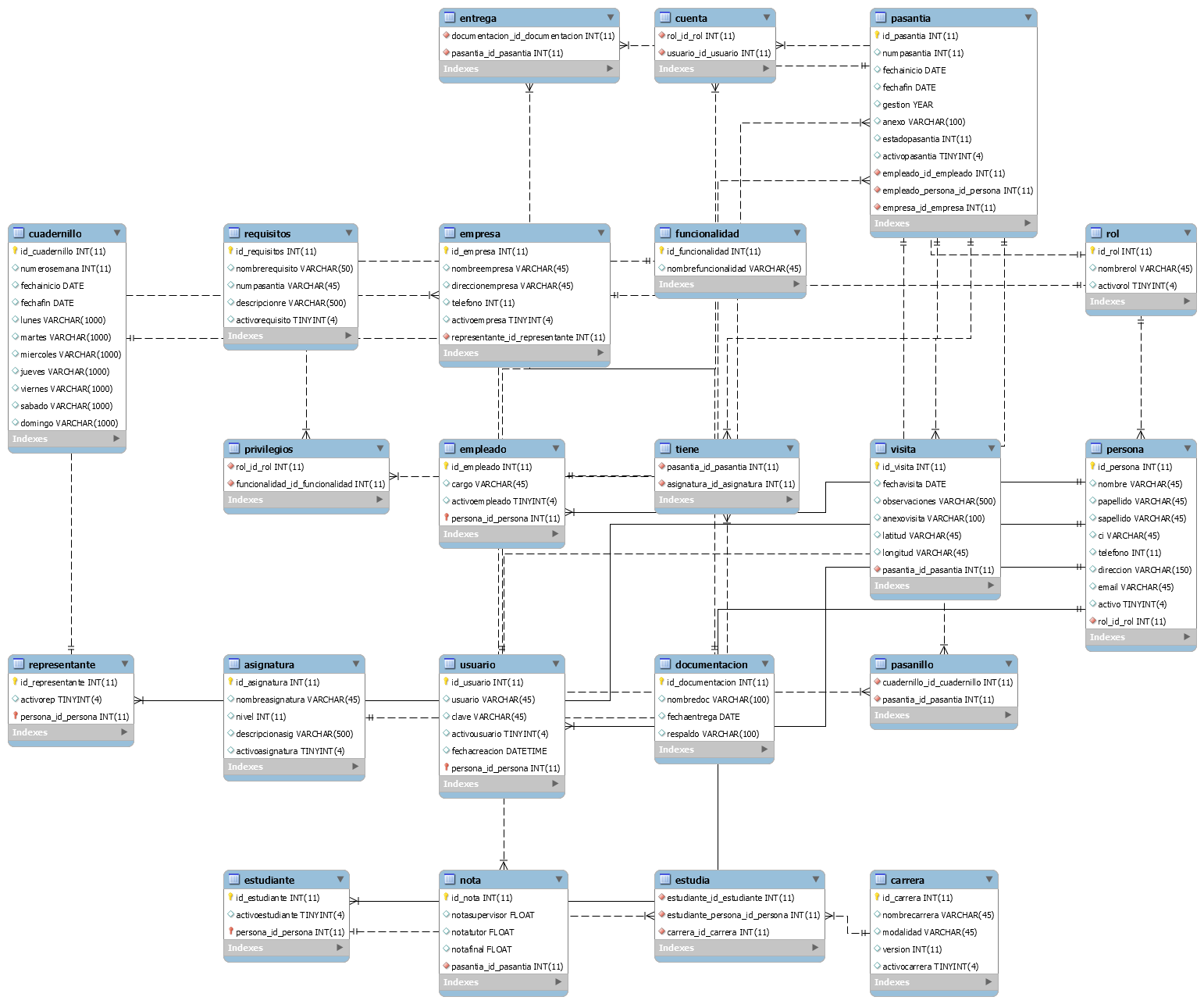
Una vez revisado los requerimientos del usuario se modifico el diagrama de clases 4 veces, donde finalmente se tiene el seguimiente diagrama de clases.

**Fig**. 24 Versión Final de la base de datos

**Ref**.:( Elaboración Propia)

## 4.3.3 **DIAGRAMA ENTIDAD RELACI**ÓN**.**

**Fig**.25 Diagrama Entidad relación

****

(**Ref**. Elaboración Propia)

# CAPITULO 5

# FASE DE IMPLEMENTACIÓN

# 5.1 INTRODUCCIÓN. -

En la etapa de prueba se verifica los resultados de la implementación probando en cada construcción de interfaz, incluyendo todos los módulos desarrollados para el sistema, y aquí es donde se realiza el control de los errores para minimizar lo mas posibles, lo cual permite asegurar que el sistema sea confiable para los usuarios.

5.2 ITERACIONES DEL SISTEMA CONTROL Y SEGUIMIENTOS DE PASANTIA**.** En la ejecución de iteraciones tenemos los siguientes

(Ref.: Fuente propia)

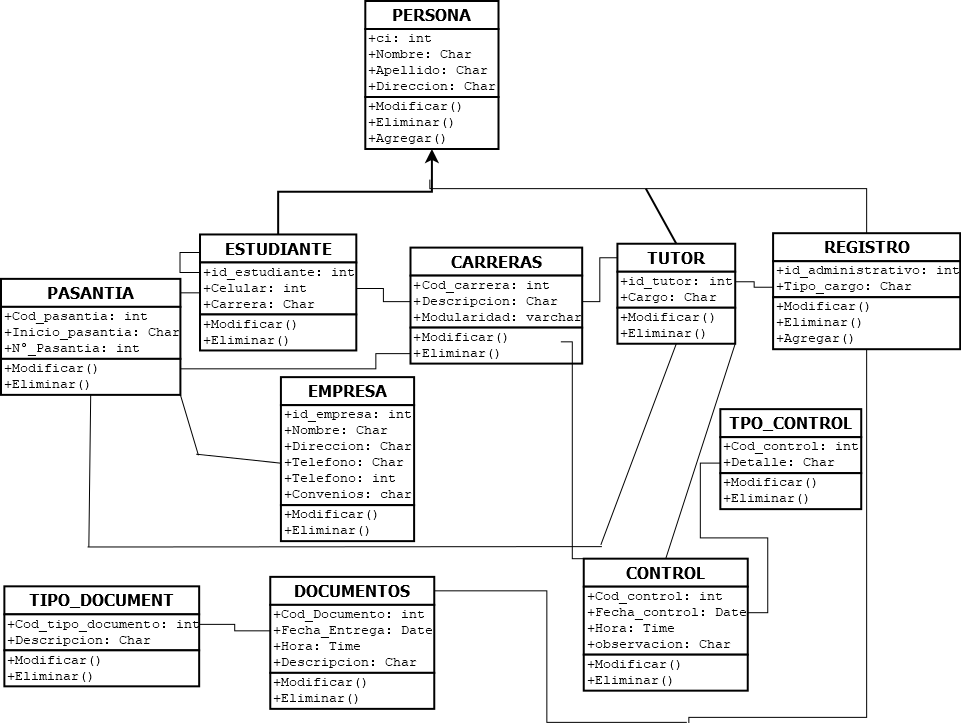
## 5.2.1 ITERACIÓN 1

Aquí se muestra el primer sprint del proceso de implementación mediante un cronograma de actividades describiendo los trabajos asignados.



### 5.2.1.1 Iteraciones hasta la primera propuesta de desarrollo y apariencia:

1. Base de Datos primera version. –Se realizo la base de datos con la ayuda de los diferentes diagramas como ser diagrama de caso de uso, clases y entidad relación.



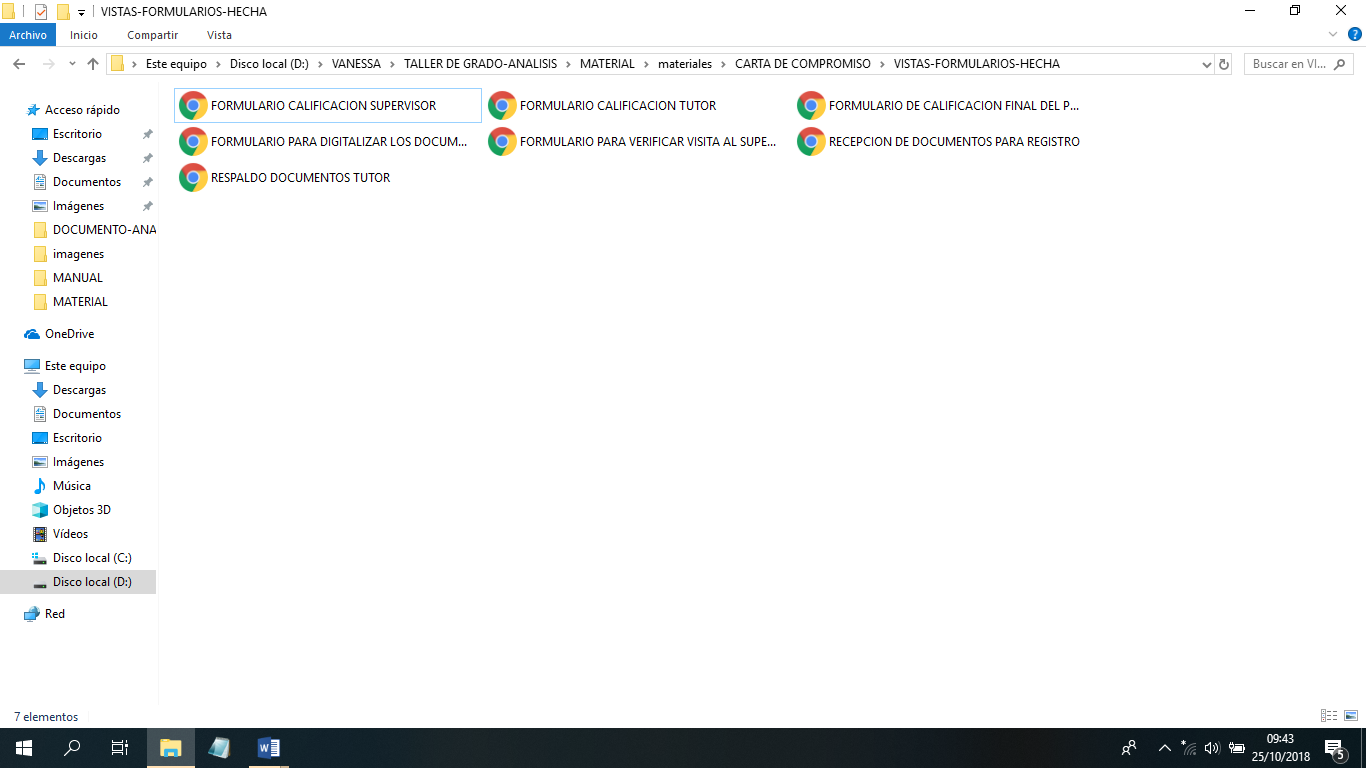
## PRIMERA VERSION

figura 26 base de datos versión 1

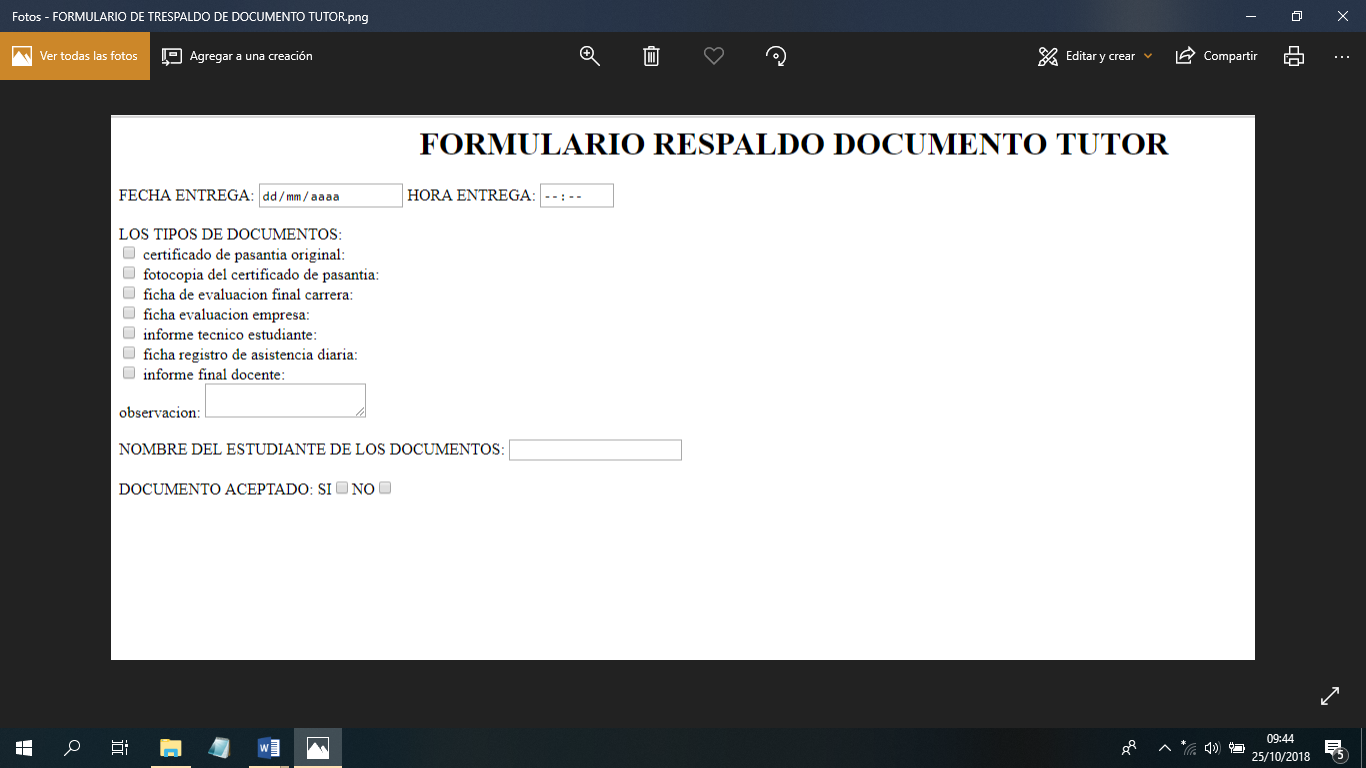
(Ref.: Fuente propia)

1. Formulario para las vistas del sistema. -Se realizo vistas previas de todas las tablas de la base de datos, con todo esto se realizó para subir los cruds al sistema.

figura 27 formularios vistas del sistema



(Ref.: Fuente propia)

**Ejemplo de los formularios:** 

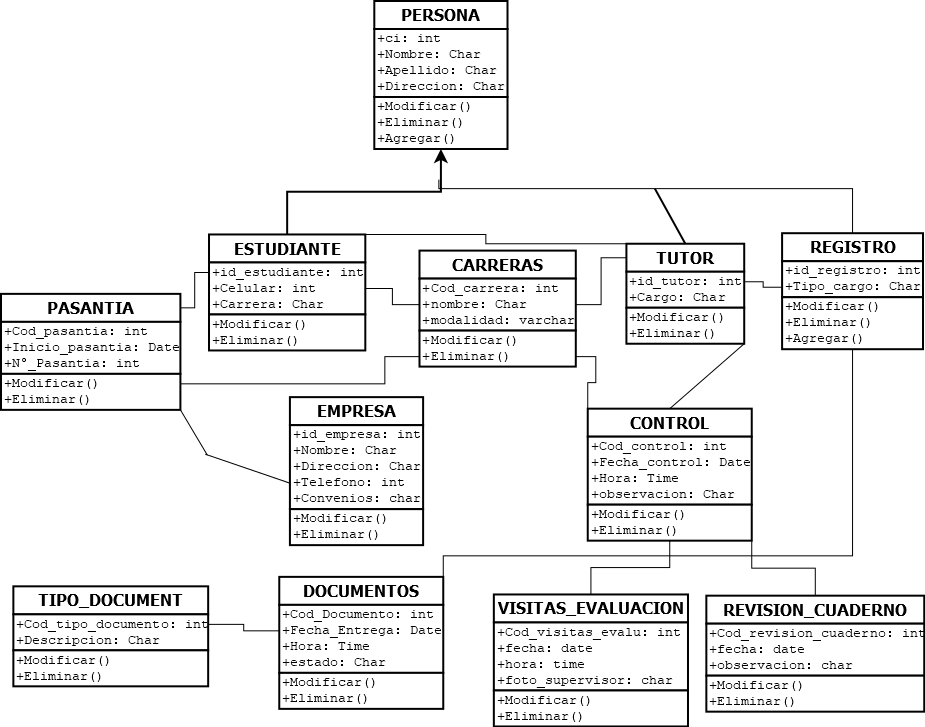
(Ref.: Fuente propia)

## ITERACIÓN 2.-

### Iteraciones hasta la segunda propuesta de desarrollo del diseño.

1. Base de datos segunda versión. –Se realizo diferentes modificaciones a lo largo de las revisiones y se llegó a una versión casi final. Que es la siguiente.

figura 28 dase de datos versión 2



(Ref.: Fuente propia)

#### Base de datos versión final. –

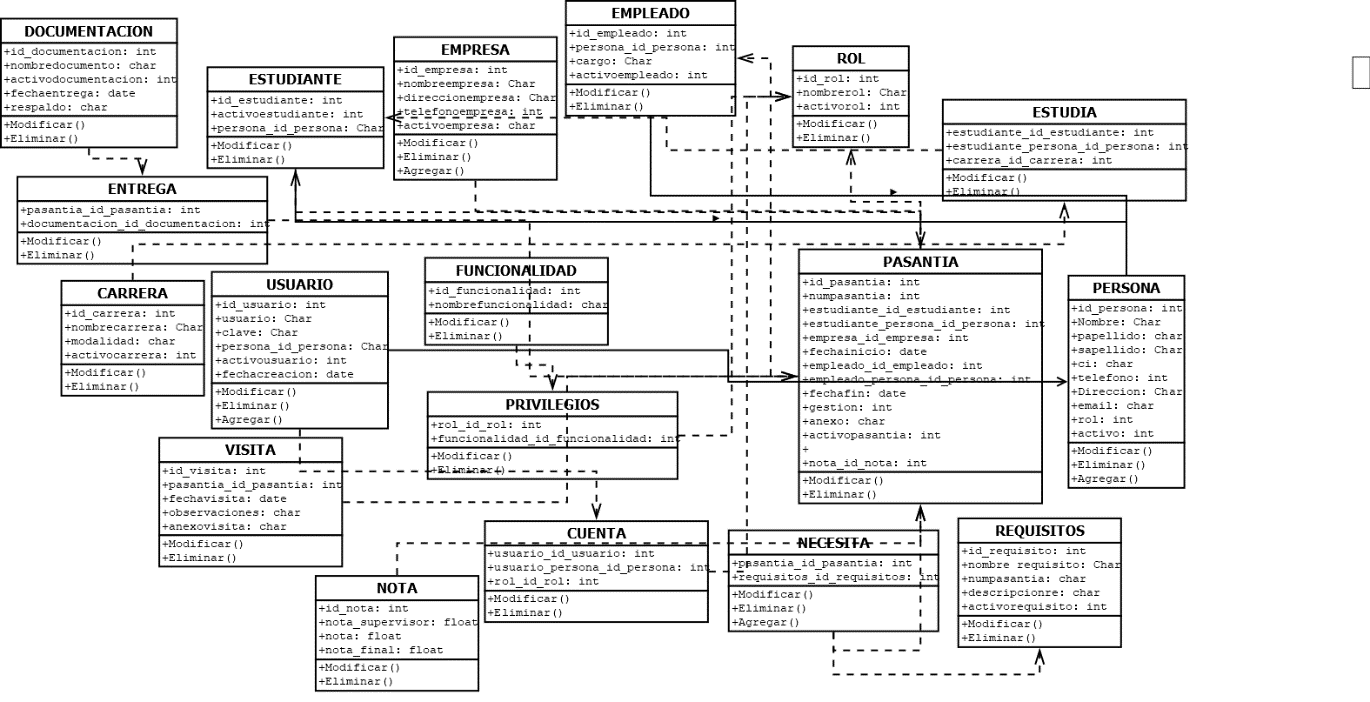


figura 29 dase de datos final

(Ref.: Fuente propia)

figura 30 Implementación de dase de datos codigo

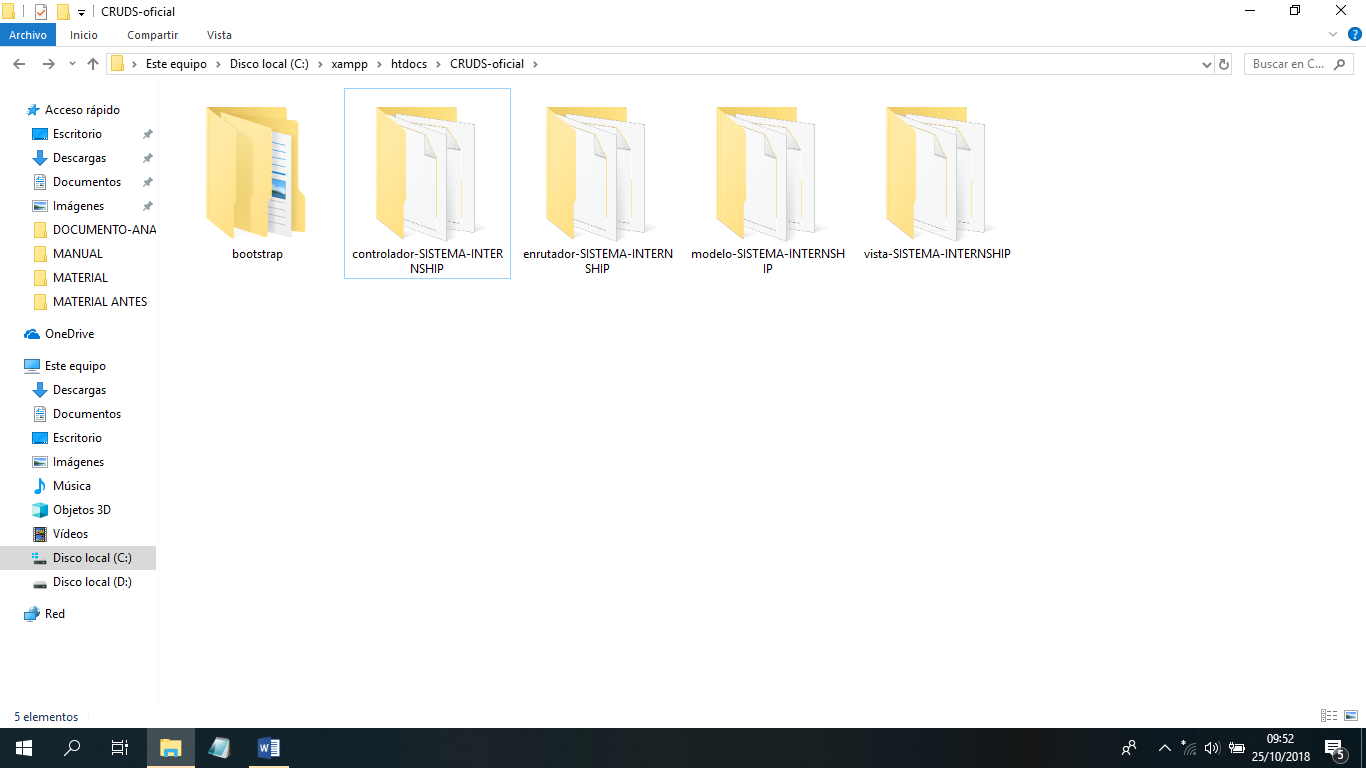


(Ref.: Fuente propia)

1. Los cruds del Sistema. **-** Se realizo cruds de cada tabla con la funcionalidad (crear, modificar, eliminar)

##### Una Vista de que se requería para los cruds.

figura 31 carpetas y vistas de lo que se requiere



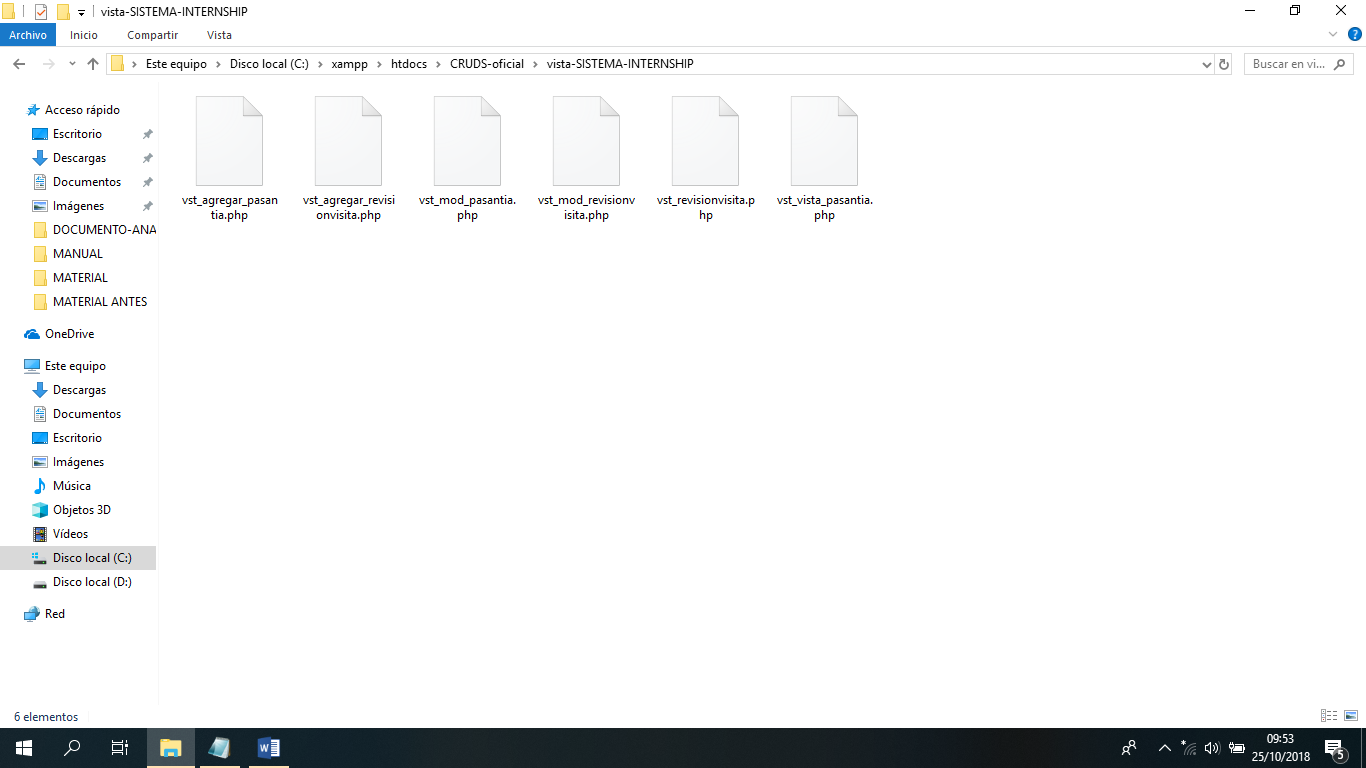
(Ref.: Fuente propia)

##### Ejemplo de las Vistas de Sistema Control y Seguimiento de Pasantias

figura 32 ejemplo de vistas de los cruds

(Ref.: Fuente propia)

(Ref.: Fuente propia)



##### Ejemplo de un cruds con funcionalidad:

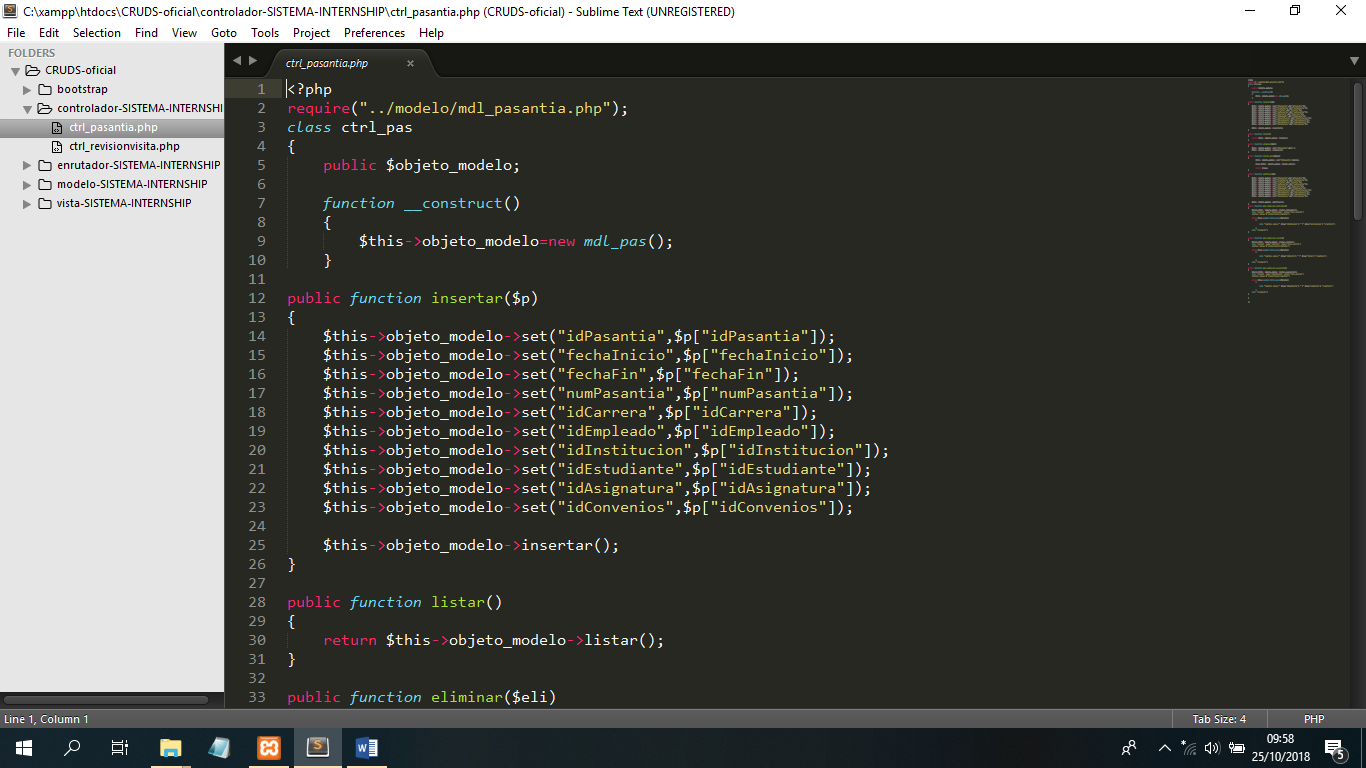
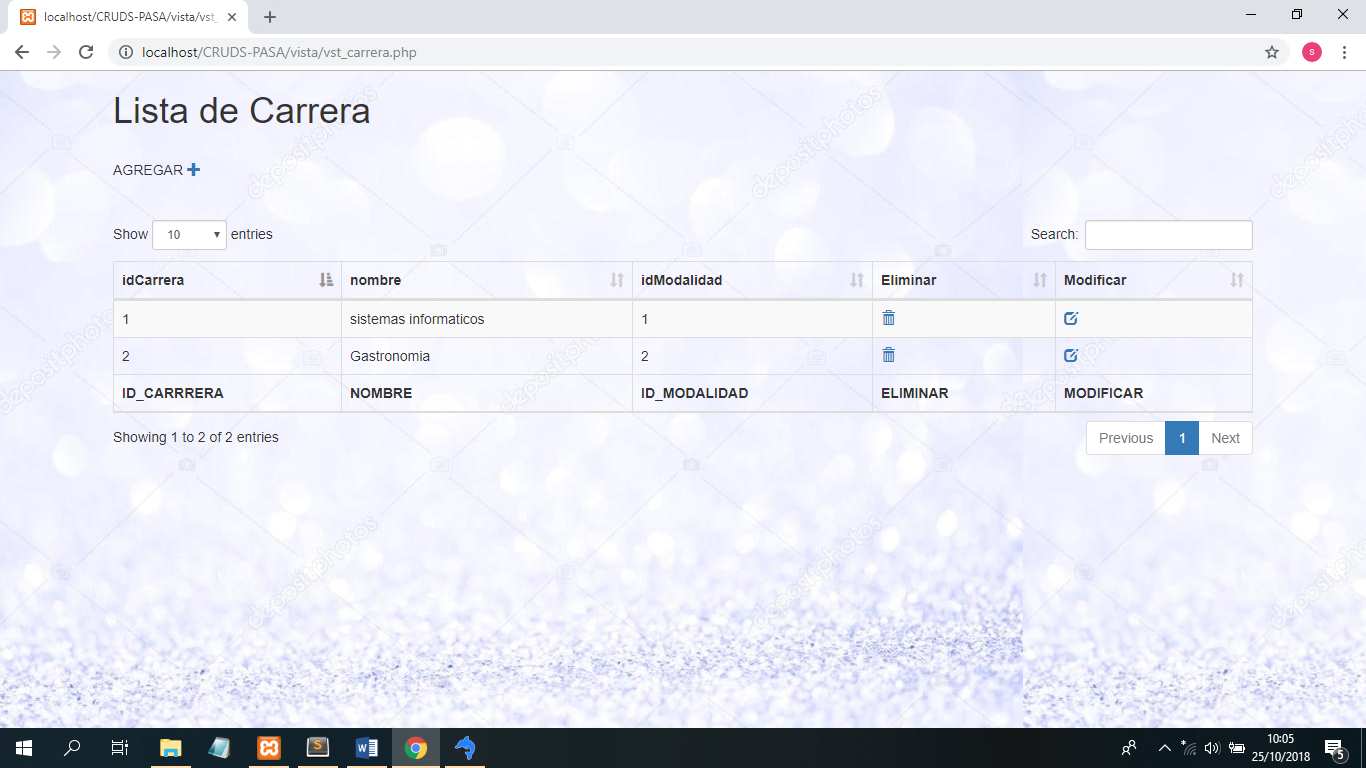


figura 33 código de los cruds

(Ref.: Fuente propia)

(Ref.: Fuente propia)

figura 34 cruds con funcionalidad



(Ref.: Fuente propia)

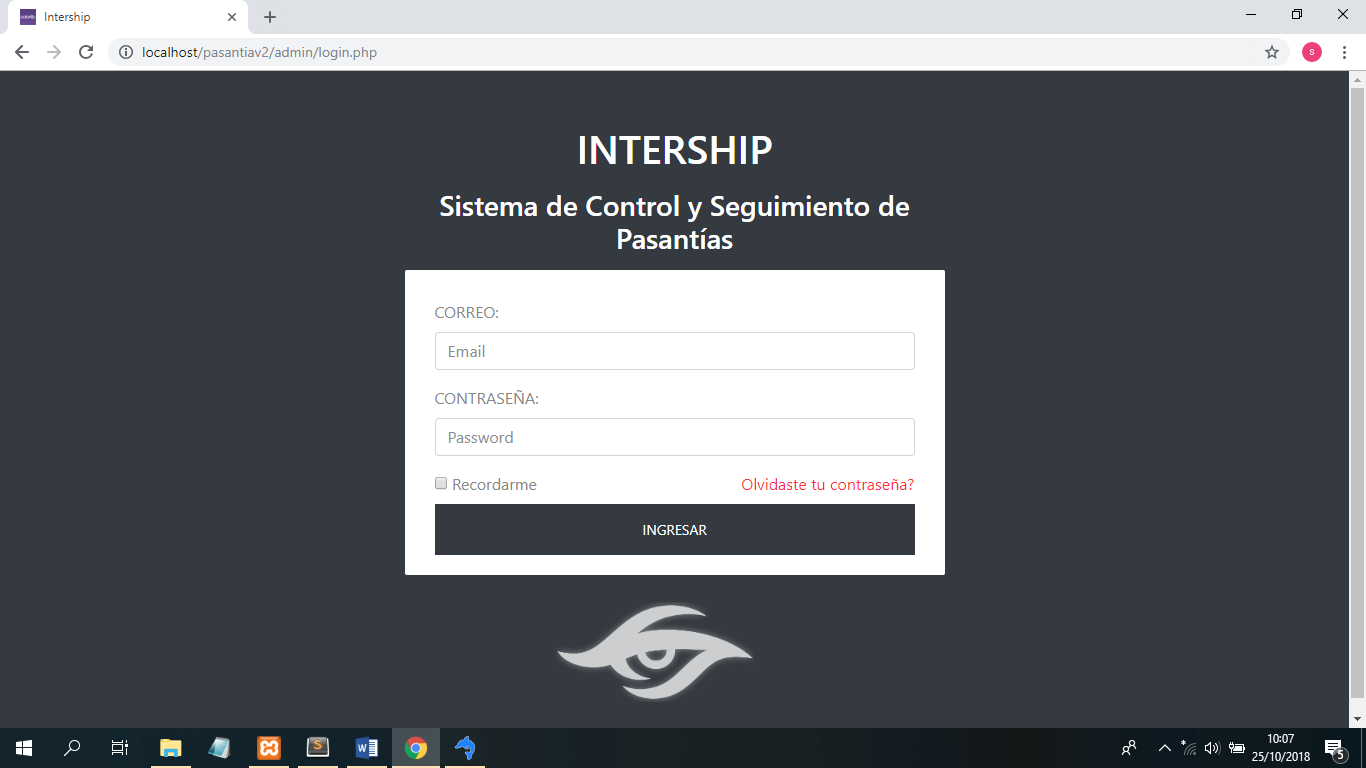
## ITERACIÓN 3.-

### A. Logueo del sistema Control y Seguimiento de Pasantías. –

**1 versión:**

figura 35 vista del sistema versión 1

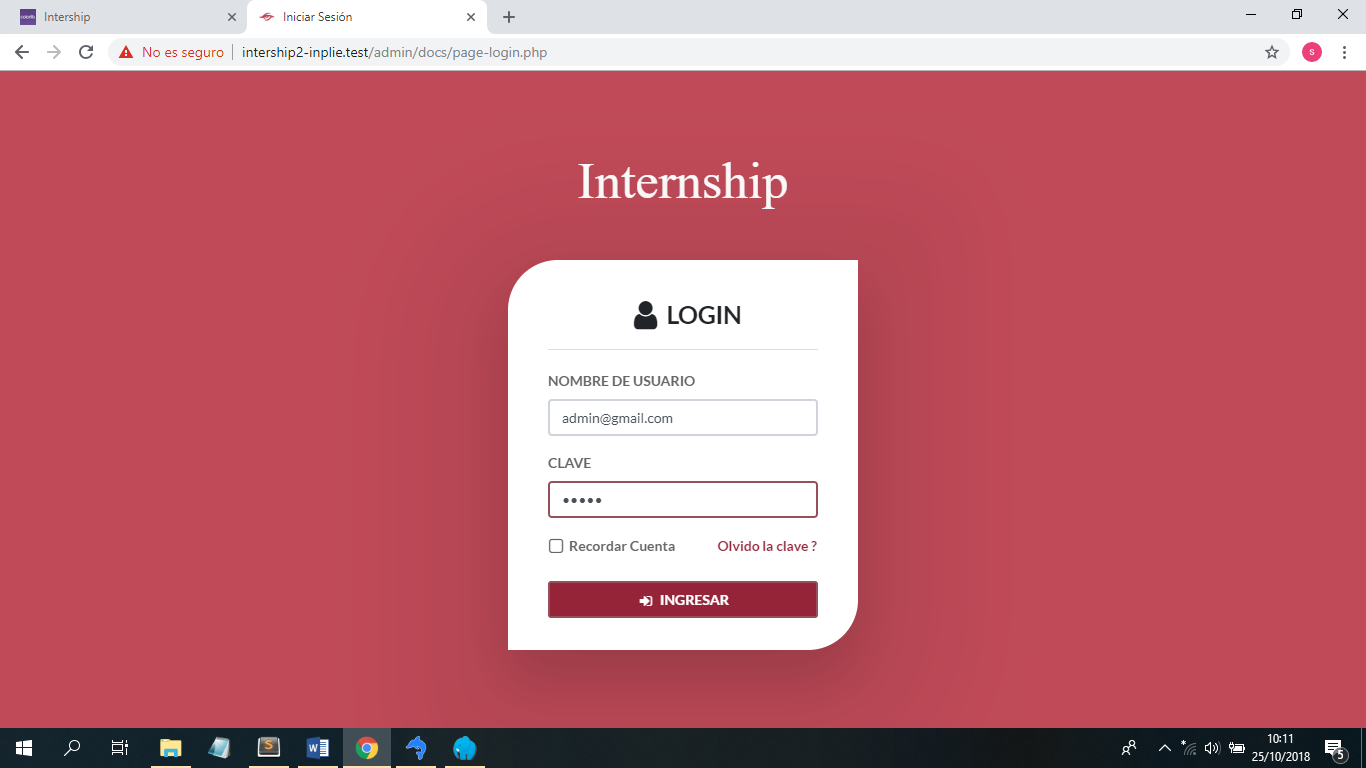
(Ref.: Fuente propia)



**Versión final:**

(Ref.: Fuente propia)

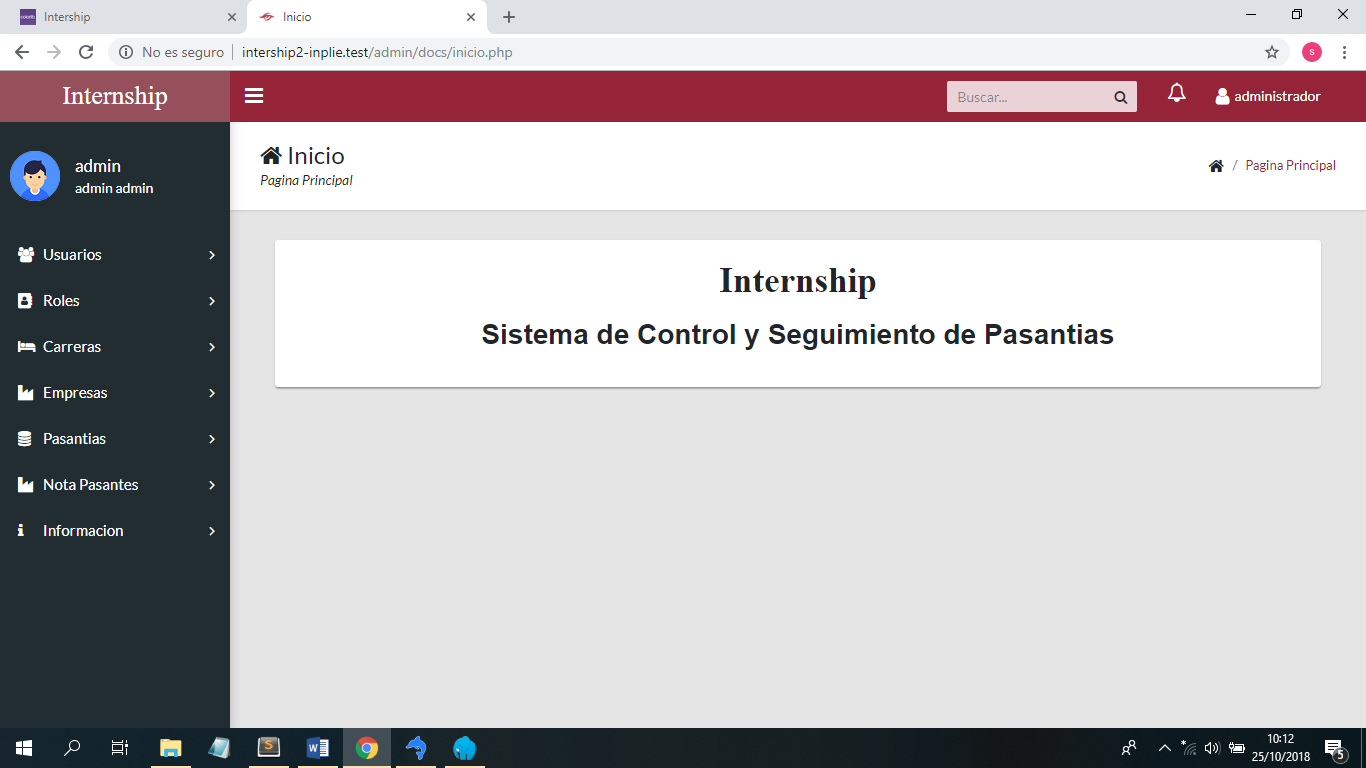
figura 36 vista del sistema versión final



**Vista oficial del sistema:**

figura 37 Vista oficial del sistema

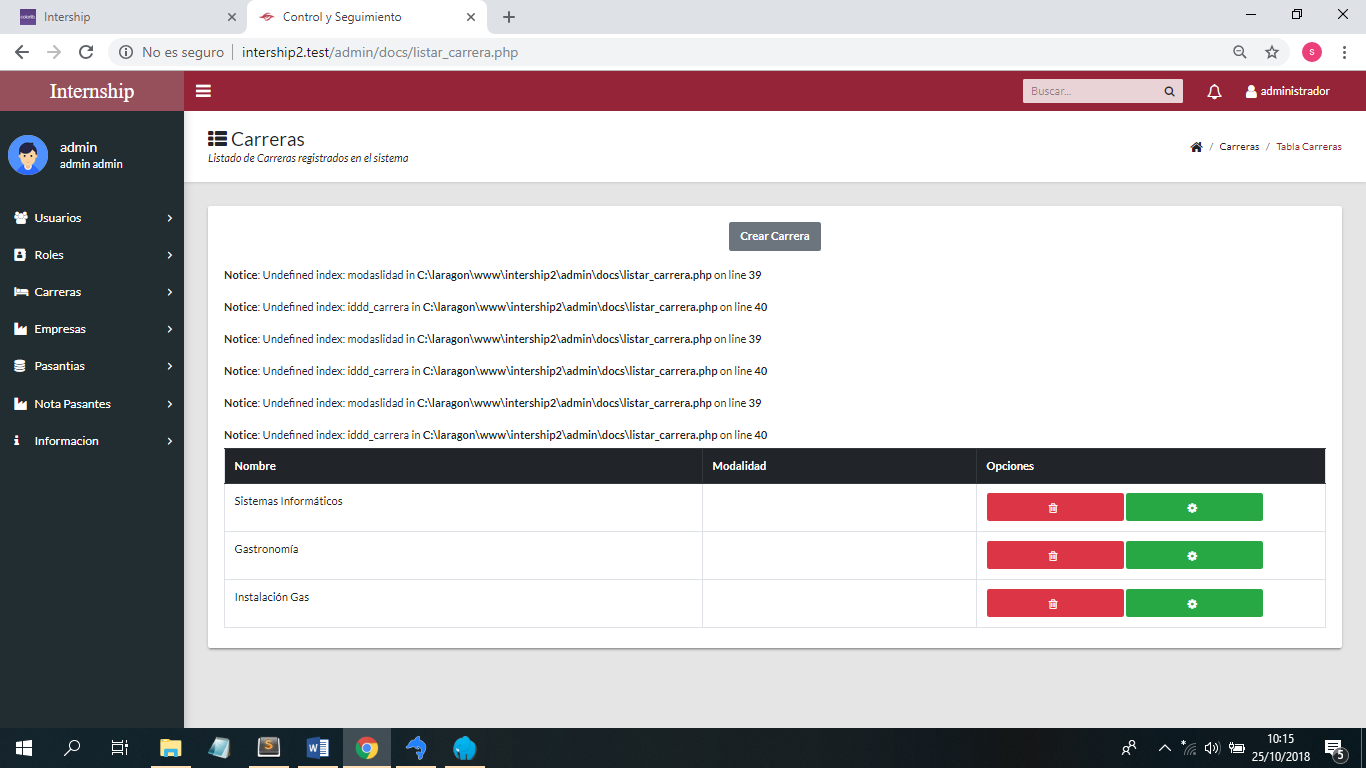
(Ref.: Fuente propia)



**Pruebas del Sistema:**

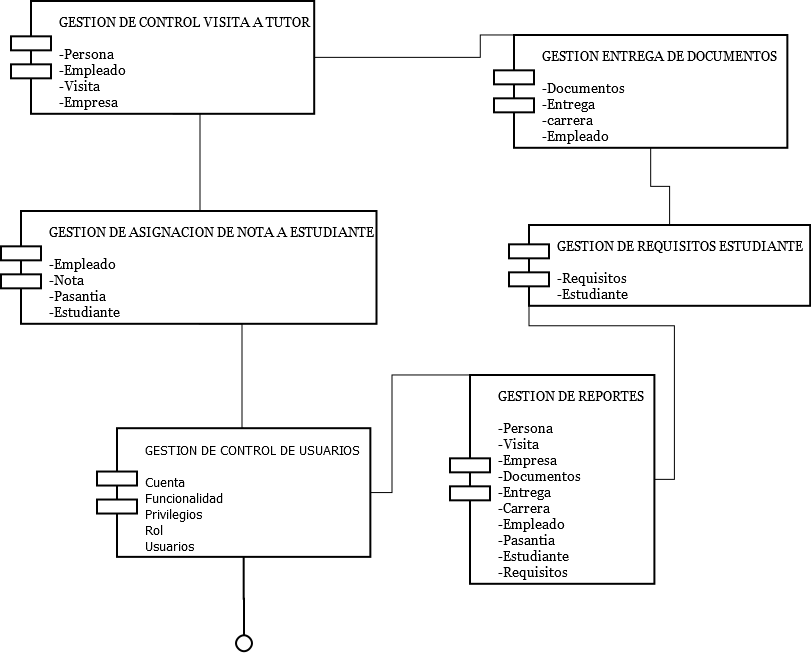
figura 38 Pruebas del Sistema

(Ref.: Fuente propia)



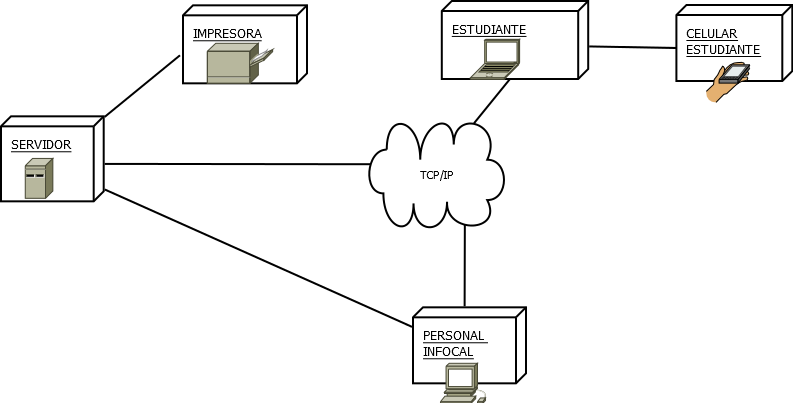
5.3 DIAGRAMA DE COMPONENTES. – Aquí se describe el diagrama de componentes del sistema INTERNSHIP.

figura 39 diagrama de componentes

****

(Ref.: Fuente propia)

5.4 DIAGRAMA DE DESPLIEGUE. - Aquí se muestra el diagrama de despliegue del sistema INTERNSHIP mostrando la interacción de los diferentes actores con el sistema de gestión de tramites

****

TCP/IP

figura 40 diagrama de despliegue

(Ref.: Fuente propia)

# 5.5 INTERFAZ DE SISTEMA

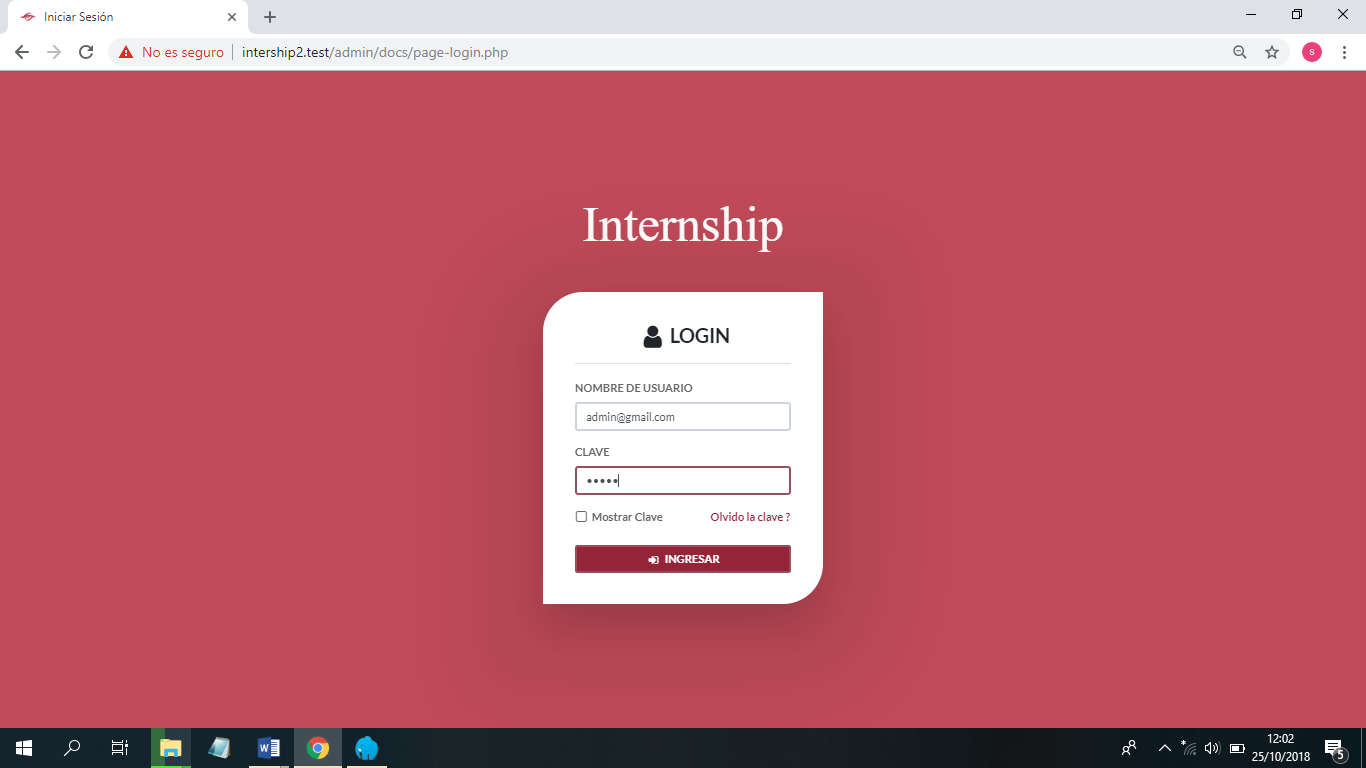
En la interfaz del sistema INTERNSHIP se muestra el principal funcionamiento describiendo las principales funciones tales como:

1. Administrador
2. Personal
3. Estudiante

## Administrador

En esta se muestra al administrador que podrá crear, modificar, visualizar todas las funcionalidades del sistema.

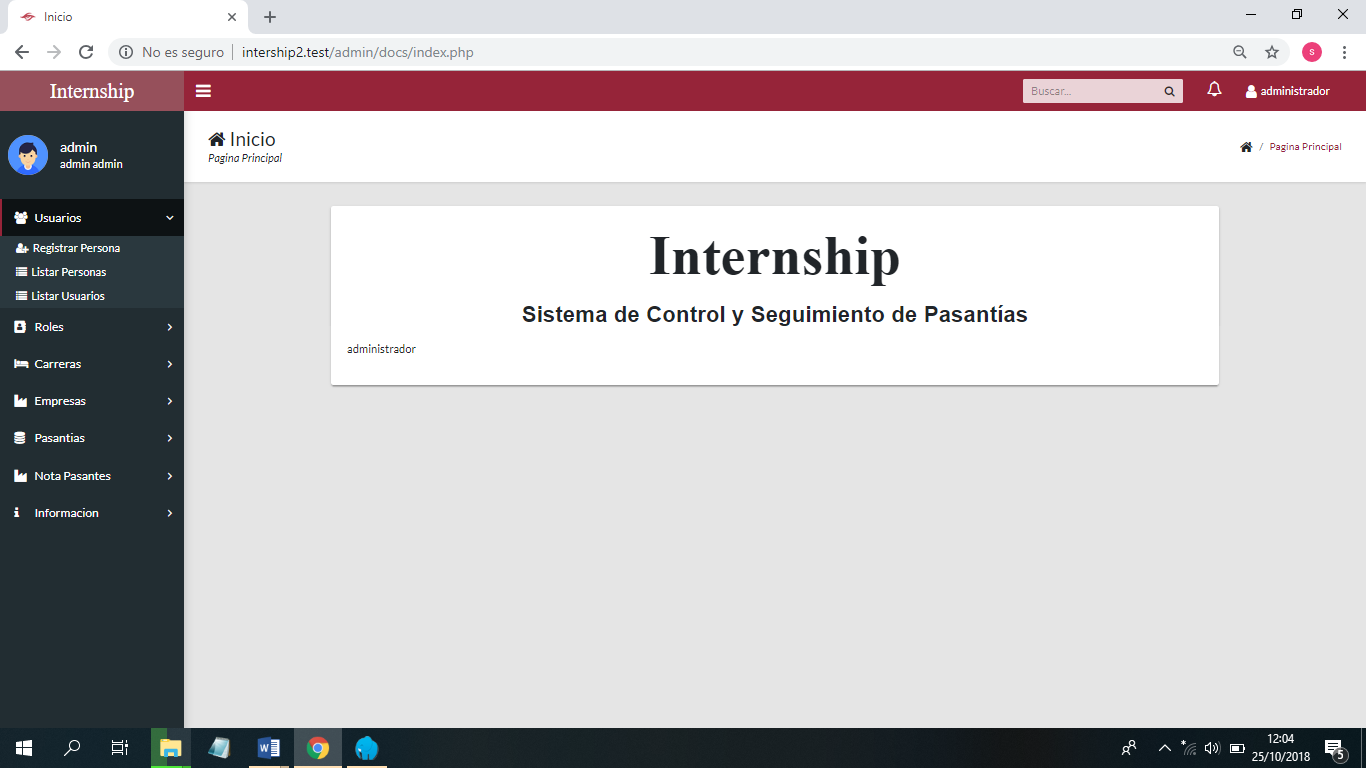
figura 41 Logueo en el sistema



(Ref.: Fuente propia)

En la figura 43 se muestra como ingresa el administrador mediante una cuenta y un usuario (Loguin).

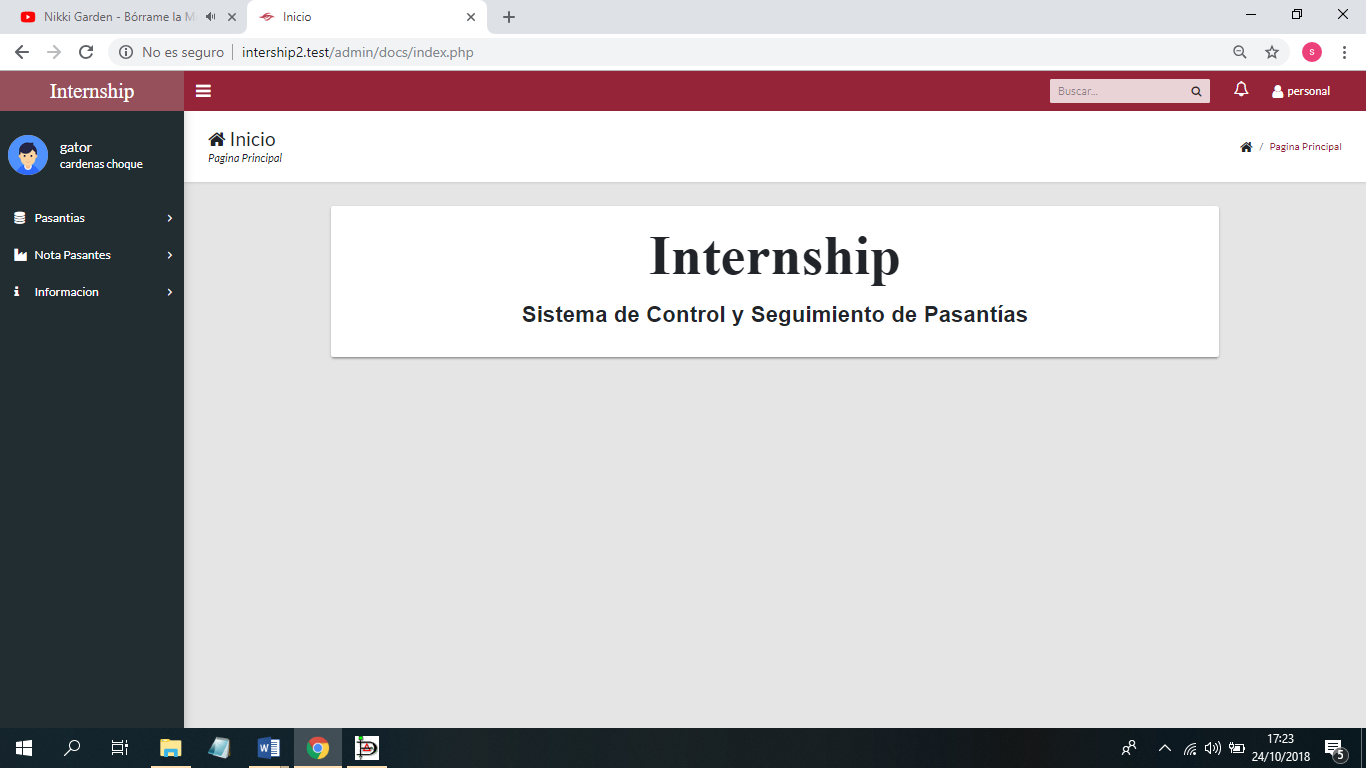
figura 42 Interfaz del Rol Administrador en el sistema INTERNSHIP



(Ref.: Fuente propia)

1. **Personal:** Es el que emitir nota de las pasantías y revisiones de cuadernillo.

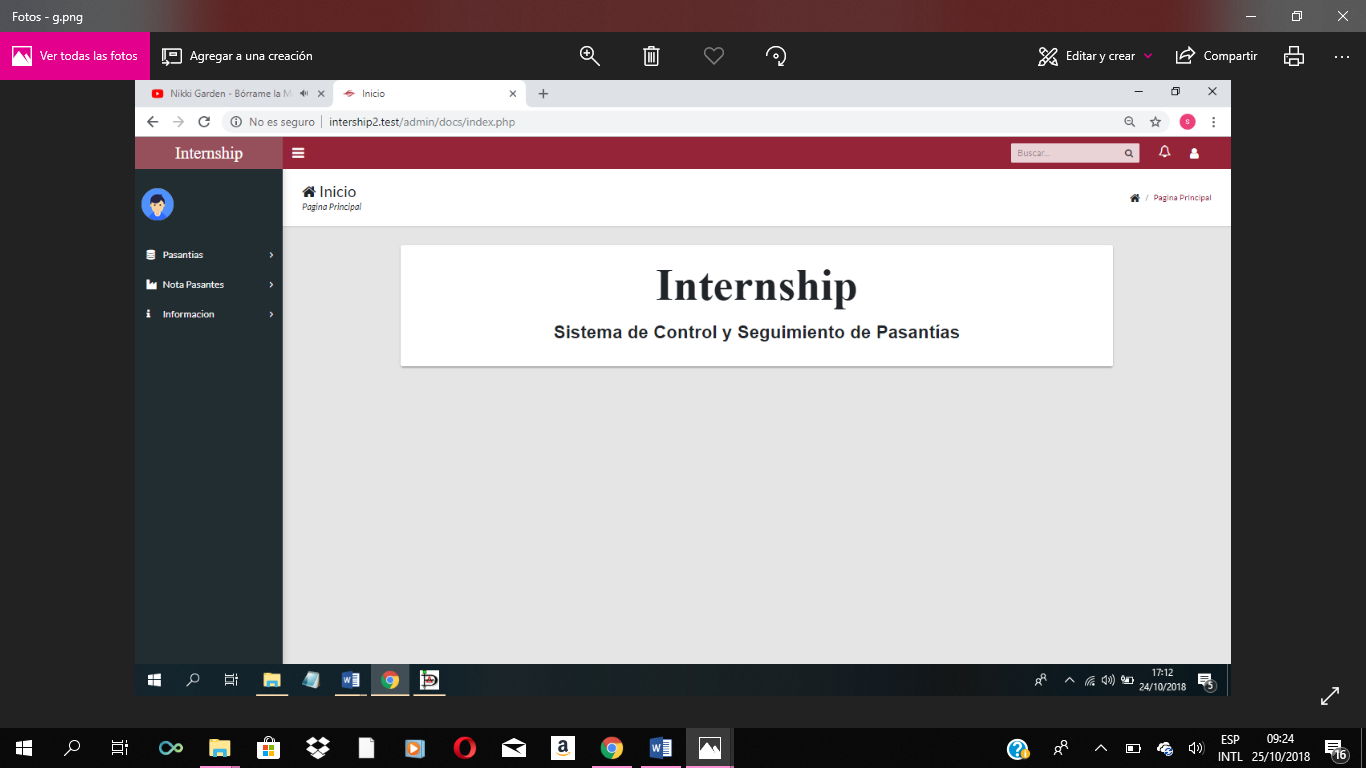
figura 43 vistas de usuario personal

****

(Ref.: Fuente propia)

1. **Estudiante:** Es el que podrá ver sus notas y ver los requisitos de la pasantía.

figura 44 vista de usuario estudiante



(Ref.: Fuente propia)

# CAPITULO 6

# CONCLUSIONES

# INTRODUCCIÓN

En la etapa de conclusión se dará un visto bueno de que porcentajes se avance en cuanto al sistema INTERNSHIP “CONTROL YSEGUIMIENTO DE PASANTIAS”, lo cual permite asegurar que el sistema sea confiable para los usuarios.

# CONCLUSIONES ACERCA DE LOS OBJETIVOS

* Se logro implementar el 99,9% del módulo control de usuarios.
* Se logro implementar el 10% de módulo de reportes, porque no alcanzo el tiempo para completar.
* Se logro implementar el 99.99% del módulo de base de datos
* Se logró implementar el 10 % del módulo de respaldo de documentos, porque no alcanzo el tiempo para completar.
* Se logró implementar 70% el Módulo de Control de pasantía.
* Se logró implementar 80% Control y Seguimiento de los estudiantes para el docente.
* Se logró implementar 50% el Control y Seguimiento de la Documentación presentada.

# CONCLUSIONES ACERCA DE LAS HERRAMIENTAS UTILIZADAS

Alinicio del proyecto se intentó utilizar el Framework Laravel, pero posteriormente se acordó en utilizar el lenguaje HTML, MySql y PHP, por el mayor conocimiento que se tenía de los mismos.

También se intentó utilizar la Herramienta Xampp para la gestión de base de datos, posteriormente se hizo oficial el uso del motor de base de datos Laragon, la cual nos ayudó a alcanzar objetivos (Mail Sender).

# CONCLUSIONES ACERCA DE LA METODOLOGÍA

La Metologia Scrum plantea un trabajo en equipo el cual fue un poco complicado de seguir por cuestiones de tiempo e ideas diversas.

Para el desarrollo del presente proyecto se determinó los siguientes roles, según Metodología SCRUM:

**Product owner🡪 (Ivan Tarqui).** Están encargado de recolectar datos mediante cuestionarios y entrevistas.

**Scrum Master🡪(Vanessa Flores)**. Están encargado de la fase de planificación y análisis, también de la división de funcionalidades y ejecución de historia de usuarios

**Front End🡪(Josué Verastegui y juan roque).** Están encargado de la fase de realizar las vistas y ventanas del sistema

**Back End 🡪(Roy Franco y Rafael Mérida**), Están encargados de la programación del sistema y de la base de datos. Tanto con los complementos extras y pruebas del sistema.

**Tester🡪(Jhonas)**. Están encargado de la fase de recolección de errores del sistema,

# CONCLUSIONES ACERCA DE LA IMPLEMENTACION

**Diseño Gráfico:**

Se utilizo Photoshop cs6 para la maqueta de diseño de la página de inicio.

**Programación:**

Se utilizó PHP puro, utilizando la arquitectura de software Modelo-Vista-Controlador (MVC), se programó en la herramienta Sublime Text ver.3.1.1.

**Base de Datos:**

Se utilizó el gestor de base de datos Xampp ver 3.2.2, después se pasó a Laragon ver. 3.1.9 y para la implementación y programación de la base de datos se utilizó MySQL WorkBench ver 6.2 y para la vista previa de la base de datos SqlYog ver.12.4.3

# RECOMENDACIONES:

1. **Recomendaciones Respecto a la Instalación del Sistema.**

El usuario debe tener en cuenta que es un software bastante intuitivo en las herramientas que posee, el teclado y mouse no requieres configuraciones especiales para su uso, como también el navegador solo necesita estar actualizado.

1. **Recomendaciones Respecto al Hardware y Software.**

Los requerimientos mínimos para el programa son los siguientes:

# REQUERIMIENTOS DE HARDWARE Y SOFTWARE.

Las características del servidor serán las siguientes.

|  |  |  |
| --- | --- | --- |
| indice | REQUERIMIENTOS DE HARDWARE Y SOFTWARE | |
| RHS1 | Disco Duro | Toshiba 10 Gb |
| RHS2 | Tarjeta Madre | MSI Gaming b85 |
| RHS3 | Procesador | Intel i3 4045 |
| RHS4 | Memoria RAM | Kingstone 2Gb |
| RHS5 | Lector DVD | LG 8x |
| RHS6 | Fuente | Generica 400W |
| RHS7 | Tarjeta de Video | Intel HD Graphics |
| RHS8 | Sistema Operativo | Windows 7 o Posterior (64bis, 32 bits) |
| RHS9 | Navegador Web | Chrome, Mozila Firefox y Opera. |

**3. Recomendaciones para el uso del sistema**

* El cliente debe tener conocimientos minimos del uso de una PC.
* Lectura del manual de Funcionamiento del sistema, misma que detalla todo el funcionamiento del sistema y de los distintos tipos de usuarios que actúan en el sistema.
* Seguir el manual de instalación al pie de la letra para el correcto funcionamiento del sistema.

**4. Recomendación para obtener reportes**

Saber correctamente el método de búsqueda para la generación de reportes por categoría.

# ANEXO

# DICCIONARIO **DE DATOS**.

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos a continuación se mostrará el diccionario de datos importado desde MYSQL.

Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información, se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos del sistema, su contenido también se emplea durante el diseño.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. **Carrera** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** | **Descripción** | | | | **Ejemplo** | | | | |  | | | |
| id\_carrera | int(11) | No |  | Clave primaria. Solamente se admiten números | Codigo de la carrera | | | | 1 | | | | |  | | | |
| nombrecarrera | varchar(50) | Sí | *NULL* | Solo admite caracteres de tipo letra | nombre de la carrera | | | | gastronomía | | | | |  | | | |
| modalidad | varchar(50) | Sí | *NULL* | Admite caracteres de tipo numero | codigo de la modalidad | | | | semestralizado | | | | |  | | | |
| activocarrera | tinyint(4) | Sí | *NULL* | BANDERA(FALSO=0/VERDADERO) | estado de la carrera | | | | 1/0 | | | | |  | | | |
| **Indices\_Carrera** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| **PRIMARY** | **BTREE** | **Sí** | **No** | **id\_carrera** | **No** | **Pertenece a la tabla carrera** | | | |
| 1. **Cuenta** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripción** | | | | | **Ejemplo** | | | |
| **usuario\_id\_usuario** | **int(11)** | **No** |  |  | **Solo admite caracteres de tipo numero** | | **codigo del usuario** | | | | | **20** | | | |
| **usuario\_persona\_id\_persona** | **int(11)** | **No** |  |  | **Solo admite caracteres de tipo letra** | | **codigo de la persona** | | | | | **30** | | | |
| **rol\_id\_rol** | **int(11)** | **No** |  | **rol -> id\_rol** | **Solo admite caracteres de tipo numero** | | **codigo del rol** | | | | | **40** | | | |
| **Indices\_Cuenta** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| **fk\_cuenta\_usuario1** | **BTREE** | **No** | **No** | **usuario\_id\_usuario** | **No** | **Pertenece a la tabla cuenta** | | | |
| **usuario\_persona\_id\_persona** | **No** |
| **fk\_cuenta\_rol1** | **BTREE** | **No** | **No** | **rol\_id\_rol** | **No** | **Pertenece a la tabla cuenta** | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** |  | | | | | |
| 1. **Documentación** | |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** | **Descripción** | | | | **Ejemplo** | | | | |  | | | |
| **id\_documentacion** | **int(11)** | **No** |  | **clave primaria. solamente se admiten numeros** | **codigo de documentacion** | | | | **50** | | | | |  | | | |
| **nombredocumento** | **varchar(100)** | **Sí** | **NULL** | **Solo admite caracteres de tipo letra** | **nombre del tipo de documento** | | | | **carta de pasantia** | | | | |  | | | |
| **activo documentación** | **tinyint (4)** | **Sí** | **NULL** | **BANDERA(FALSO=0/VERDADERO)** | **estado del documento** | | | | **1/0** | | | | |  | | | |
| **fechaentrega** | **datetime** | 1. **Sí** | **NULL** | **Solo admite de tipo date** | **fecha de entrega del documento** | | | | **02/05/2018** | | | | |  | | | |
| **respaldo** | **varchar(150)** | **Sí** | **NULL** | **Solo admite solamente de tipo letra** | **respado del documento** | | | | **doc.docx** | | | | |  | | | |
| **Índices\_Documentacion** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| **PRIMARY** | **BTREE** | **Sí** | **No** | **id\_documentacion** | **No** | **Pertenece a la tabla documentación** | | | |
| 1. **Empleado** | |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripción** | | | | | **Ejemplo** | | | |
| **id\_empleado** | **int(11)** | **No** |  |  | **CLAVE PRIMARIA.SOLAMENTE SE ADMITEN NUMEROS** | | **codigo del empleado** | | | | | **60** | | | |
| **persona\_id\_persona** | **int(11)** | **No** |  | **persona -> id\_persona** | **Solo admite caracteres de tipo letra** | | **codigo de la persona** | | | | | **30** | | | |
| **cargo** | **varchar(45)** | **Sí** | **NULL** |  | **Solo admite caracteres de tipo letra** | | **cargo del empleado** | | | | | **tutor** | | | |
| **activoempleado** | **tinyint(4)** | **Sí** | **NULL** |  | **BANDERA(FALSO=0/VERDADERO)** | |  | | | | | | | **estado del empleado** | 1/0 | | | |
| **Índices\_ Empleado** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_empleado | No | Pertenece a la tabla empleado | | | |
| persona\_id\_persona | No |
| fk\_empleado\_persona1 | BTREE | No | No | persona\_id\_persona | No | Pertenece a la tabla empleado | | | |
| 1. **Empresa** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Descripcion** | **Ejemplo** | |  | | | | | | |  |  | | | |
| id\_empresa | int(11) | No |  | codigo de la empresa | 70 | |  | | | | | | |  |  | | | |
| nombreempresa | varchar(50) | Sí | *NULL* | nombre de la empresa | comteco | |  | | | | | | |  |  | | | |
| direccionempresa | varchar(100) | Sí | *NULL* | dirección de la empresa | centro/villazon | |  | | | | | | |  |  | | | |
| telefonoempresa | int(11) | Sí | *NULL* | telefono de la empresa | 258745 | |  | | | | | | |  |  | | | |
| activoempresa | varchar(45) | Sí | *NULL* | estado de la empresa | 1/0 | |  | | | | | | |  |  | | | |
|  |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Índices\_Empresa** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_empresa | No | Pertenece a la tabla empresa | | | |
| 1. **Entrega** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | | **Ejemplo** | | | |
| pasantia\_id\_pasantia | int(11) | No |  | pasantia -> id\_pasantia | CLAVE PRIMARIA.SOLAMENTE SE ADMITEN NUMEROS | | codigo pasantia | | | | | 80 | | | |
| documentacion\_id\_documentacion | int(11) | No |  | documentacion -> id\_documentacion | Solamente admite de tipo numero | | codigo documentacion | | | | | 50 | | | |
| **Índices\_Entrega** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| fk\_entrega\_pasantia1 | BTREE | No | No | pasantia\_id\_pasantia | No | Pertenece a la tabla entrega | | | |
| fk\_entrega\_documentacion1 | BTREE | No | No | documentacion\_id\_documentacion | No | Pertenece a la tabla entrega | | | |
| 1. **Estudia** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | | **Ejemplo** | | | |
| estudiante\_id\_estudiante | int(11) | No |  |  | Clave primaria. Solamente se admiten números | | codigo estudiante | | | | | 90 | | | |
| estudiante\_persona\_id\_persona | int(11) | No |  |  | Solo admite de tipo letra | | codigo persona | | | | | 30 | | | |
| carrera\_id\_carrera | int(11) | No |  | carrera -> id\_carrera | Solo admite de tipo letra | | codigo carrera | | | | | 1 | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Cardinalidad** | | **Nulo** | | | | | **Comentario** | | | |
| **Índices\_ Estudia** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| fk\_estudia\_estudiante1 | BTREE | No | No | estudiante\_id\_estudiante | No | Pertenece a la tabla estudiante | | | |
| estudiante\_persona\_id\_persona | No |
| fk\_estudia\_carrera1 | BTREE | No | No | carrera\_id\_carrera | No | Pertenece a la tabla carrera | | | |
| 1. **Estudiante** | |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripción** | | | | | **Ejemplo** | | | |
| id\_estudiante | int(11) | No |  |  | Clave primaria. Solamente se admiten números | | codigo estudiante | | | | | 90 | | | |
| activoestudiante | tinyint(4) | Sí | *NULL* |  | BANDERA(FALSO=0/VERDADERO) | | estado del estudiante | | | | | 0/1 | | | |
| persona\_id\_persona | int(11) | No |  | persona -> id\_persona | Solo admite de tipo letra | | codigo persona | | | | | 30 | | | |
| **Índices\_ Estudiante** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_estudiante | No | Pertenece a la tabla estudiante | | | |
| persona\_id\_persona | No |
| fk\_estudiante\_persona1 | BTREE | No | No | persona\_id\_persona | No | Pertenece a la tabla persona | | | |
| 1. **Funcionalidad** | |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** | **Descripcion** | | | | **Ejemplo** | | | | |  | | | |
| id\_funcionalidad | int(11) | No |  | Clave primaria. Solamente se admiten números | codigo de la funcionalidad | | | | 100 | | | | |  | | | |
| nombrefuncionalidad | varchar(50) | Sí | *NULL* | Solo admite de tipo letra | nombre de la funcionalidad | | | | crear/editar | | | | |  | | | |
| **Índices\_ Funcionalidad** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_funcionalidad | No | Pertenece a la tabla funcionalidad | | | |
| 1. **Necesita** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | | **Ejemplo** | | | |
| pasantia\_id\_pasantia | int(11) | No |  | pasantia -> id\_pasantia | Clave primaria. Solamente se admiten números | | codigo de la pasantia | | | | | 110 | | | |
| requisitos\_id\_requisitos | int(11) | No |  | requisitos -> id\_requisitos | Solo admite de tipo numero | | codigo de los requisitos | | | | | 120 | | | |
| **Índices\_ Necesita** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| fk\_nesecita\_pasantia1 | BTREE | No | No | pasantia\_id\_pasantia | No | Pertenece a la tabla pasantía | | | |
| fk\_nesecita\_requisitos1 | BTREE | No | No | requisitos\_id\_requisitos | No | Pertenece a la tabla requisitos | | | |
| 1. **Nota** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** | **Descripcion** | | | | **Ejemplo** | | | | |  | | | |
| id\_nota | int(11) | No |  | Clave primaria. Solamente se admiten números | codigo nota | | | | 130 | | | | |  | | | |
| notasupervisor | float | Sí | *NULL* | Solamente admite de tipo numero | nota final supervisor | | | | 85 | | | | |  | | | |
| nota | float | Sí | *NULL* | Solamente admite de tipo numero | nota del pasante | | | | 85 | | | | |  | | | |
| notafinal | float | Sí | *NULL* | Solamente admite de tipo numero | nota final | | | | 85 | | | | |  | | | |
| **Índices\_ Nota** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_nota | No | Pertenece a la tabla nota | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** |  |  | | | |
|  |  |  |  |  |  |  | | | |
| 1. **Pasantía** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | | **Ejemplo** | | | |
| id\_pasantia | int(11) | No |  |  | Clave primaria. Solamente se admiten números | | codigo de la pasantia | | | | | 110 | | | |
| numpasantia | int(4) | Sí | *NULL* |  | Solamente admite de tipo numero | |  | | | | | III | | | |
| estudiante\_id\_estudiante | int(11) | No |  |  | Solamente admite de tipo numero | | numero de pasantia | | | | | 90 | | | |
| estudiante\_persona\_id\_persona | int(11) | No |  |  | Solamente admite de tipo numero | | codigo persona | | | | | 30 | | | |
| empresa\_id\_empresa | int(11) | No |  | empresa -> id\_empresa | Solamente admite de tipo numero | | codigo empresa | | | | | 70 | | | |
| fechainicio | timestamp | Sí | *NULL* |  | Solamente admite de tipo date | | fecha inicio de la pasantia | | | | | 02/03/2018 | | | |
| empleado\_id\_empleado | int(11) | No |  |  | Solamente admite de tipo numero | | codigo del empleado | | | | | 60 | | | |
| empleado\_persona\_id\_persona | int(11) | No |  |  | Solamente admite de tipo numero | | codigo de la persona | | | | | 30 | | | |
| fechafin | datetime | Sí | *NULL* |  | Solamente admite de tipo date | | fecha final de la pasantia | | | | | 10/08/2018 | | | |
| gestion | yar(4) | Sí | *NULL* |  | Solamente admite de tipo letra | | año de la pasantia | | | | | 2018 | | | |
| anexo | varchar(50) | Sí | *NULL* |  | Solamente admite de tipo letra | | anexo de la pasantia | | | | | ANEXO.DOCX | | | |
| activopasantia | tinyint(4) | Sí | *NULL* |  | BANDERA(FALSO=0/VERDADERO) | | estado de la pasantia | | | | | 1/0 | | | |
| nota\_id\_nota | int(11) | No |  | nota -> id\_nota | Solamente admite de tipo numero | | codigo de la nota | | | | | 130 | | | |
| **Índices\_ Pasantía** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_pasantia | No | Pertenece a la tabla pasantia | | | |
| fk\_pasantia\_estudiante1 | BTREE | No | No | estudiante\_id\_estudiante | No | Pertenece a la tabla estudiante | | | |
| estudiante\_persona\_id\_persona | No |
| fk\_pasantia\_empresa1 | BTREE | No | No | empresa\_id\_empresa | No | Pertenece a la tabla empresa | | | |
| fk\_pasantia\_empleado1 | BTREE | No | No | empleado\_id\_empleado | No | Pertenece a la tabla empleado | | | |
| empleado\_persona\_id\_persona | No |
| fk\_pasantia\_nota1 | BTREE | No | No | nota\_id\_nota | No | Pertenece a la tabla nota | | | |
| 1. **Persona** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Descripcion** | | **Ejemplo** | | | | | | |  |  | | | |
| id\_persona | int(11) | No |  |  | codigo de la persona | | 30 | | | | | | |  |  | | | |
| nombre | varchar(50) | Sí | *NULL* |  | nombre de la persona | | vanessa | | | | | | |  |  | | | |
| papellido | varchar(50) | Sí | *NULL* |  | primer apellido | | flores | | | | | | |  |  | | | |
| sapellido | varchar(50) | Sí | *NULL* |  | segundo apellido | | mamni | | | | | | |  |  | | | |
| ci | varchar(10) | Sí | *NULL* |  | numero de carnet de la persona | | 9399952 | | | | | | |  |  | | | |
| telefono | int(11) | Sí | *NULL* |  | telefono de la persona | | 5884755 | | | | | | |  |  | | | |
| direccion | varchar(100) | Sí | *NULL* |  | direccion de la persona | | sacaba | | | | | | |  |  | | | |
| email | varchar(100) | Sí | *NULL* |  | correo de la persona | | [vane24@gmail.com](mailto:vane24@gmail.com) | | | | | | |  |  | | | |
| rol | int(11) | Sí | *NULL* | rol -> id\_rol | cargo de la persona | | 140 | | | | | | |  |  | | | |
| activo | tinyint(4) | Sí | *NULL* |  | estado de la persona | | 0/1 | | | | | | |  |  | | | |
|  |  |  |  |  |  | |  | | | | | | |  |  | | | |
|  |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Índices\_ Persona** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_persona | No | Pertenece a la tabla persona | | | |
| fk\_persona\_rol | BTREE | No | No | rol | Sí | Pertenece a la tabla rol | | | |
| 1. **Privilegios** | |  |  |  |  | |  | | | |  | | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | **Ejemplo** | | | | |
| rol\_id\_rol | int(11) | No |  | rol -> id\_rol | Clave primaria. Solamente se admiten números | | codigo del rol | | | | 140 | | | | |
| funcionalidad\_id\_funcionalidad | int(11) | No |  | funcionalidad -> id\_funcionalidad | Solo admite de tipo numero | | codigo de la funcionalidad | | | | 100 | | | | |
| **Índices\_ Privilegios** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| fk\_privilegios\_rol1 | BTREE | No | No | rol\_id\_rol | No | Pertenece a la tabla rol | | | |
| fk\_privilegios\_funcionalidad1 | BTREE | No | No | funcionalidad\_id\_funcionalidad | No | Pertenece a la tabla | | | |
| 1. **Requisitos** | |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** | **Descripcion** | | | **Ejemplo** | | | | |  | | | |
| id\_requisitos | int(11) | No |  | Clave primaria. Solamente se admiten números | codigo de los requisitos | | | 120 | | | | |  | | | |
| nombrerequisito | varchar(50) | Sí | *NULL* | Solo admiten de tipo numero | nombre de los requisitos | | | entrega de documentos | | | | |  | | | |
| numpasantia | varchar(45) | Sí | *NULL* | Solo admiten de tipo letra | numero de pasantia | | | III | | | | |  | | | |
| descripcionre | varchar(500) | Sí | *NULL* | Solo admiten de tipo letra | descripcion requisito | | | aun esta vigente | | | | |  | | | |
| activorequisito | tinyint(4) | Sí | *NULL* | BANDERA(FALSO=0/VERDADERO) | estado delrequisito | | | 0/1 | | | | |  | | | |
| **Índices\_ Requisitos** |  |  |  |  |  | | |  | | | | |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_requisitos | No | Pertenece a la tabla requisitos | | | |
| 1. **Rol** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Comentarios** | **Descripcion** | | | **Ejemplo** | | | | |  | | | |
| id\_rol | int(11) | No |  | Clave primaria. Solamente se admiten números | codigo rol | | | 140 | | | | |  | | | |
| nombrerol | varchar(50) | Sí | *NULL* | Solo admiten de tipo letra | nombre de rol | | | administrador | | | | |  | | | |
| activorol | tinyint(4) | Sí | *NULL* | BANDERA(FALSO=0/VERDADERO) | estado del rol | | | 0/1 | | | | |  | | | |
| **Índices\_ Rol** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_rol | No | Pertenece a la tabla rol | | | |
| 1. **Usuario** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | **Ejemplo** | | | | |
| id\_usuario | int(11) | No |  |  | Clave primaria. Solamente se admiten números | | codigo usuario | | | | 150 | | | | |
| usuario | varchar(50) | Sí | *NULL* |  | Solo admiten de tipo letra | | nombre del usuario | | | | [vane24@gmail.com](mailto:vane24@gmail.com) | | | | |
|  |  |  |  |  |  | | contraseña del usuario | | | | vane123 | | | | |
| clave | varchar(50) | Sí | *NULL* |  | Solo admiten de tipo letra | |
| persona\_id\_persona | int(11) | No |  | persona -> id\_persona | Solo admiten de tipo numero | | codigo de la persona | | | | 30 | | | | |
| activousuario | tinyint(4) | Sí | *NULL* |  | BANDERA(FALSO=0/VERDADERO) | | estado usuario | | | | 0/1 | | | | |
| fechacreacion | datetime | Sí | *NULL* |  | Solo admiten de tipo date | | fecha de creacion de usuario | | | | 01/01/2018 | | | | |
| **Índices\_ Usuario** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_usuario | No | Pertenece a la tabla usuario | | | |
| persona\_id\_persona | No |
| fk\_usuario\_persona1 | BTREE | No | No | persona\_id\_persona | No | Pertenece a la tabla persona | | | |
| 1. **Visita** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Columna** | **Tipo** | **Nulo** | **Predeterminado** | **Enlaces a** | **Comentarios** | | **Descripcion** | | | | **Ejemplo** | | | | |
| id\_visita | int(11) | No |  |  | Clave primaria. Solamente se admiten números | | codigo visita | | | | 160 | | | | |
| pasantia\_id\_pasantia | int(11) | No |  | pasantia -> id\_pasantia | Solo admiten de tipo numero | | codigo pasantia | | | | 110 | | | | |
| fechavisita | datetime | Sí | *NULL* |  | Solo admiten de tipo date | | fecha de visita | | | | 08/06/2018 | | | | |
| observaciones | varchar(500) | Sí | *NULL* |  | Solo admiten de tipo letra | | observacion de la visita | | | | todo excelente | | | | |
| anexovisita | varchar(100) | Sí | *NULL* |  | Solo admiten de tipo letra | | anexo de la visita | | | | hecho | | | | |
| **Índices\_ Visita** |  |  |  |  |  | |  | | | | | | |  |  | | | |
| **Nombre de la clave** | **Tipo** | **Único** | **Empaquetado** | **Columna** | **Nulo** | **Comentario** | | | |
| PRIMARY | BTREE | Sí | No | id\_visita | No | Pertenece a la tabla visita | | | |
| fk\_visita\_pasantia1 | BTREE | No | No | pasantia\_id\_pasantia | No | Pertenece a la tabla visita pasantía | | | |

# GLOSARIO DE TERMINOS

**Pasantía:** Practica profesional que realiza un estudiante para poner en práctica sus conocimientos y facultades. El pasante es el aprendiz que lleva adelante esta práctica con la intención de obtener experiencia de campo mientras que el encargado mientras que el encargado que lo guía se lo conoce como tutor.

**Tutor:** Es quien acompañara, aconsejara, corregir, estimular, cuestionar y supervisar cada uno de los pasos del pasante a lo largo de sus practicas

**Ficha de inicio pasantía:** es la ficha que llena el pasante para dar inicio a su pasantía en la empresa, aquí se detallan los objetivos de la pasantía, fechas de inicio y conclusión, tareas a realizar, obligaciones a cumplir, etc.

**Ficha de evaluación**. – La Ficha de Monitoreo y Evaluación es un instrumento de evaluación y monitoreo que permite mostrar el avance de los programas federales de desarrollo social de forma estructurada, sintética y homogénea para un ejercicio fiscal, con el objetivo de contribuir a la toma de decisiones y mejora de los programas y acciones.

**Supervisión.** - Vigilancia o dirección de la realización de una actividad determinada por parte de una persona con autoridad o capacidad para ello.

**Requisitos.** - es una circunstancia o condición necesaria para algo. Puede emplearse en muy diversos ámbitos. Como ser en este caso requisitos que se necesita para tomar la pasantía que necesariamente se requiere para seguir con el siguiente paso.

**Control.** - Examen u observación. Seguimiento del pasante en correlación a la empresa o institución que asiste, tanto docente y estudiante son controlados para un mejor y eficiente seguimiento.

**Cronograma.** - Representación gráfica de un conjunto de hechos en función del tiempo del proyecto o sistema que esté realizando.

**Recolección.** - Acción o actividad de recolectar. Recolección de información, para el sistema, documentación y nuevas modificaciones que se realicen en el proyecto.

**Asignación.** – Asignar nuevas tareas, asignar campos en la documentación y nuevas historias de usuario para el sistema.

**Recepciona**. - Recibir archivos o documentos de determinadas personas de la institución y verificar su estado, nuevas disposiciones para la modificación en el sistema y nueva recepción de información para el documento.

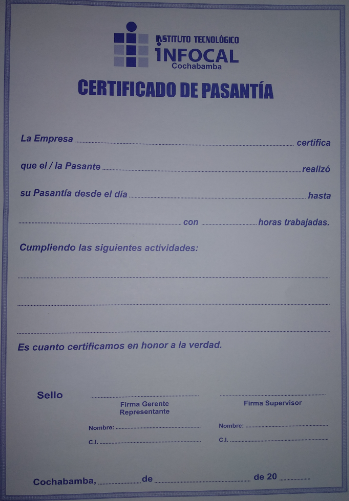
**Carta de pasantía. -** Se entiende como carta de pasante, a un documento que es entregado por una institución para constar o dar fe del trabajo que se está realizando o que se ha realizado.

**Visita:** Es el seguimiento físico del tutor de la pasantía en la empresa para evaluar el trabajo del pasante.

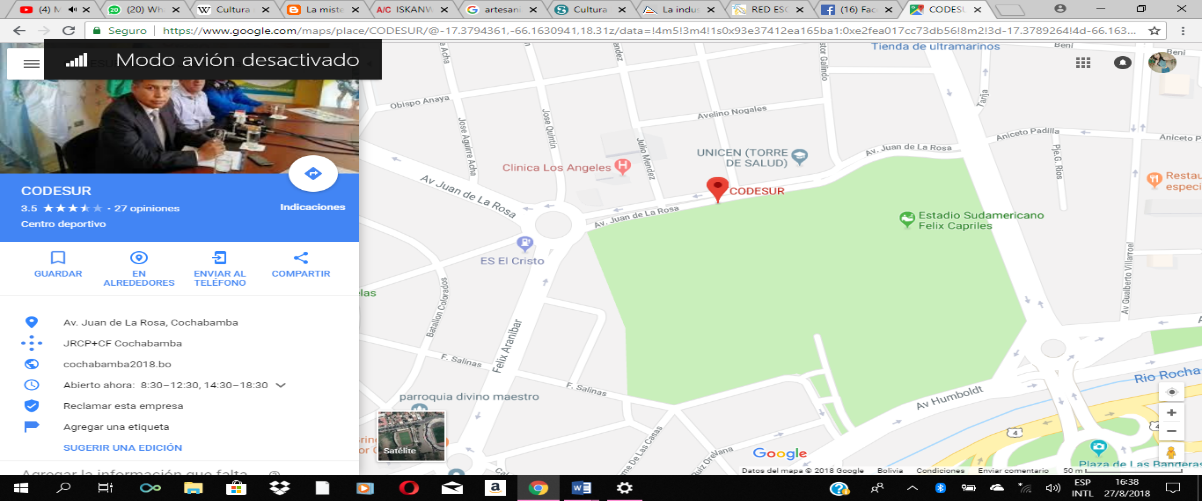
**Fecha de entrega:** Es la fecha en la que se entrega el informe de todo lo que hizo en su pasantía y a ver destacado todo lo aprendido.

# DOCUMENTACION DE PASANTIAS

## Certificado de pasantía.

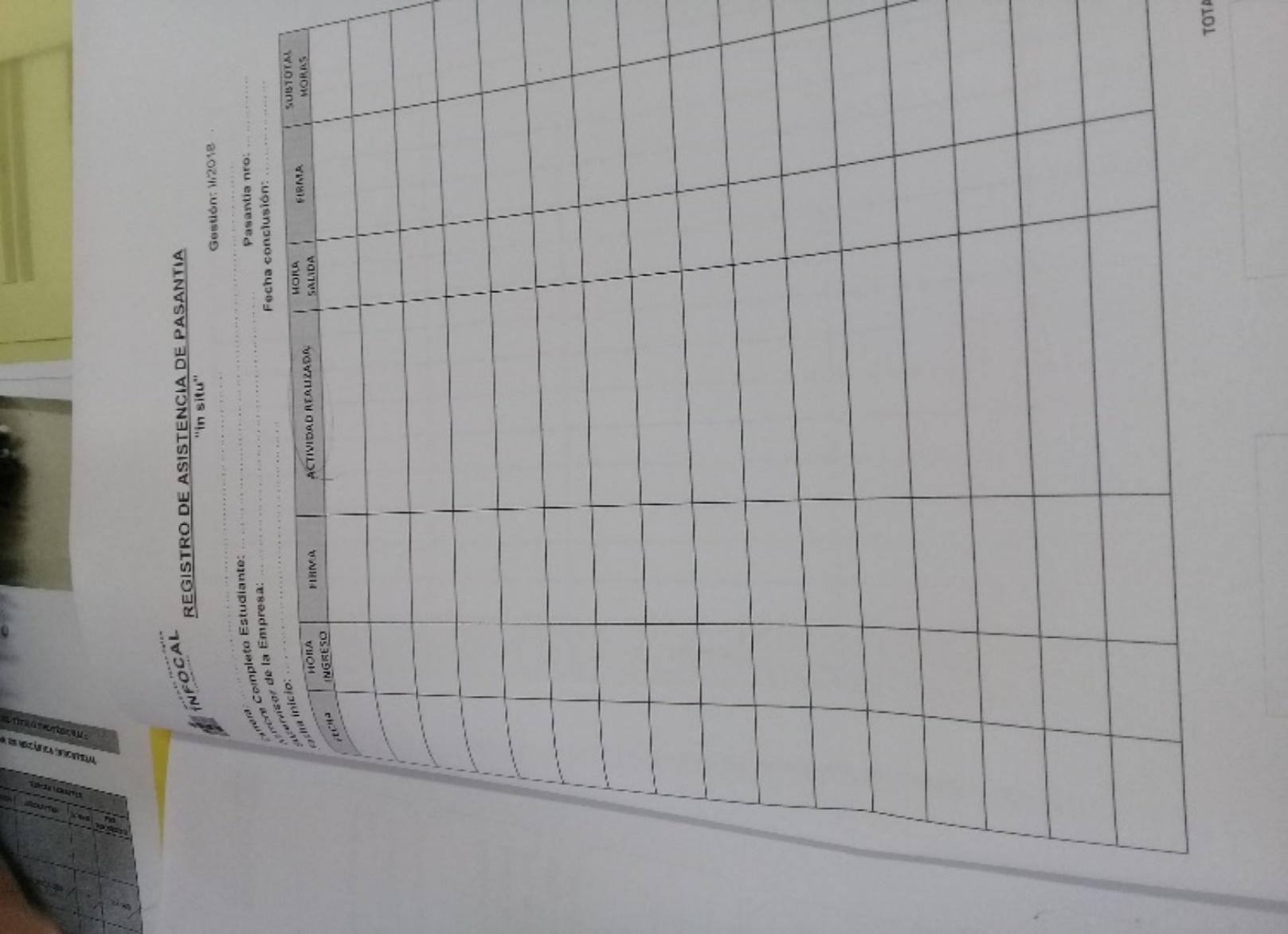


## CROQUIS DE EMPRESA

****En esta parte tenemos que poner un croquis y la ubicación exacta donde queda la empresa en la cual realizaste tu pasantía Ejem.

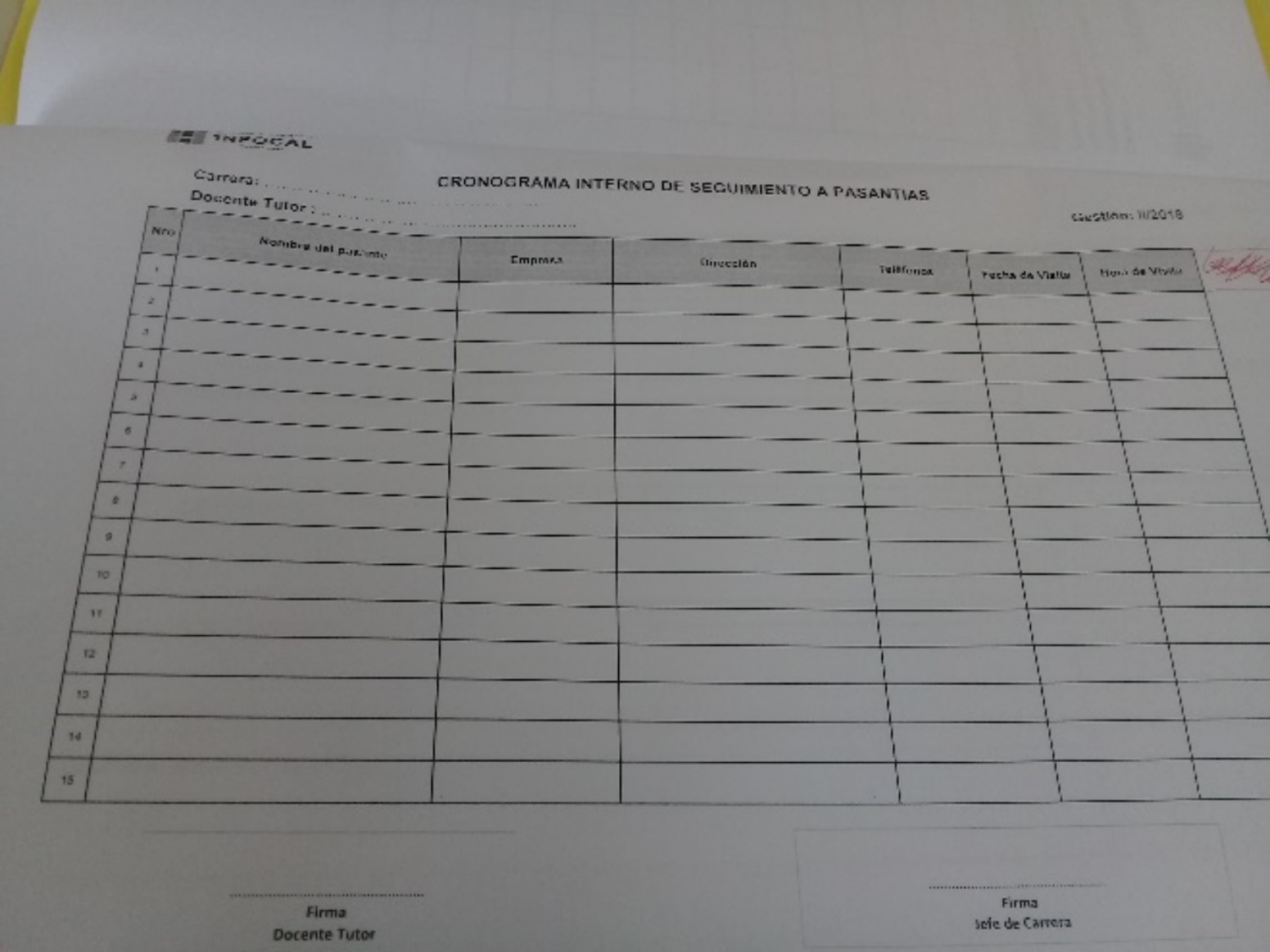
# FICHA DE PLAN DE TRABAJO CARRERA

# CONTROL DE EVALUACION:



# FICHA DE REGISTRO DE ASISTENCIA DE PASANTIA

# CRONOGRAMA INTERNO DE SEGUIMIENTO DE PASANTIA



1. <http://www.spw.cl/proyectos/apuntes2/cap_6.htm> (VISITA: 08-sep-2018) [↑](#footnote-ref-1)
2. <https://okhosting.com/blog/el-ciclo-de-vida-del-software/> (VISITA: 08-sep-2018) [↑](#footnote-ref-2)
3. <https://proyectosagiles.org/desarrollo-iterativo-incremental/> (VISITA: 08-sep-2018) [↑](#footnote-ref-3)
4. <https://www.lucidchart.com/pages/es/qu%C3%A9-es-el-lenguaje-unificado-de-modelado-uml> (VISITA: 08-sep-2018) [↑](#footnote-ref-4)
5. <https://www.monografias.com/trabajos94/modelado-sistemas-uml/modelado-sistemas-uml.shtml> (VISITA: 08-sep-2018) [↑](#footnote-ref-5)
6. <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml> (VISITA: 08-sep-2018) [↑](#footnote-ref-6)
7. <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-componentes/> (VISITA: 08-sep-2018) [↑](#footnote-ref-7)
8. <http://ingsoftware-luiszambrano.blogspot.com/2015/07/diagrama-de-despliegue.html> (VISITA: 08-sep-2018) [↑](#footnote-ref-8)
9. <https://es.wikipedia.org/wiki/PHP>(VISITA: 08-sep-2018) [↑](#footnote-ref-9)
10. <https://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/> (VISITA: 08-sep-2018) [↑](#footnote-ref-10)
11. <http://myu-charly.blogspot.com/> (VISITA: 08-sep-2018) [↑](#footnote-ref-11)
12. <https://es.wikipedia.org/wiki/Microsoft_Visio> (VISITA: 08-sep-2018) [↑](#footnote-ref-12)
13. [https://www.ecured.cu/SQLyog (VISITA: 08-sep-2018)](https://www.ecured.cu/SQLyog%2030/VII/2018) [↑](#footnote-ref-13)
14. https://es.scribd.com/doc/173514397/PARA-QUE-SIRVE-MYSQL-WORKBENCH-docx. (VISITA: 08-sep-2018) [↑](#footnote-ref-14)
15. [https://es.wikipedia.org/wiki/Sublime\_Text (VISITA: 08-sep-2018)](https://es.wikipedia.org/wiki/Sublime_Text%2030/VII/2018) [↑](#footnote-ref-15)
16. <https://es.wikipedia.org/wiki/Adobe_Photoshop> (VISITA: 08-sep-2018) [↑](#footnote-ref-16)
17. https://es.scribd.com/doc/173514397/PARA-QUE-SIRVE-MYSQL-WORKBENCH-docx.(VISITA:08-sep-2018) [↑](#footnote-ref-17)
18. [https://es.wikipedia.org/wiki/Sublime\_Text (VISITA: 08-sep-2018)](https://es.wikipedia.org/wiki/Sublime_Text%2030/VII/2018) [↑](#footnote-ref-18)
19. <https://es.wikipedia.org/wiki/Adobe_Photoshop> (VISITA: 08-sep-2018) [↑](#footnote-ref-19)
20. <https://es.wikipedia.org/wiki/Google_Chrome> (VISITA: 08-sep-2018) [↑](#footnote-ref-20)