

**Título:** Aplicación web para la gestión del proceso de la Guardia

Obrera Estudiantil en la facultad de Matemática, Física y Computación.

**Autor:** Fidel Rojo Pérez

**Tutor:** Dr. C. Daniel Gálvez Lio

Este documento es Propiedad Patrimonial de la Universidad Central “Marta Abreu” de Las Villas, y se encuentra depositado en los fondos de la Biblioteca Universitaria “Chiqui Gómez Lubian” subordinada a la Dirección de Información Científico Técnica de la mencionada casa de altos estudios.

Se autoriza su utilización bajo la licencia siguiente: **Atribución- No Comercial- Compartir Igual**



Para cualquier información contacte con:

Dirección de Información Científico Técnica. Universidad Central “Marta Abreu” de Las

Villas. Carretera a Camajuaní. Km 5½. Santa Clara. Villa Clara. Cuba. CP. 54 830 Teléfonos.: +53 01 42281503-1419

# **Dedicatoria**

# **Agradecimientos**

# **Resumen**

# **Abstract**

# **Tabla de contenido**

[**Dedicatoria** iii](#_Toc114754979)

[**Agradecimientos** iv](#_Toc114754980)

[**Resumen** v](#_Toc114754981)

[**Abstract** vi](#_Toc114754982)

[**Tabla de contenido** vii](#_Toc114754983)

[**Tabla de ilustraciones** ix](#_Toc114754984)

[**Introducción** 1](#_Toc114754985)

[**Problema Investigación** 2](#_Toc114754986)

[**Objetivo General** 2](#_Toc114754987)

[**Objetivos específicos** 2](#_Toc114754988)

[**Preguntas** 2](#_Toc114754989)

[**Justificación** 3](#_Toc114754990)

[**Estructura del documento** 3](#_Toc114754991)

[**Capítulo I. Fundamentación teórica** 4](#_Toc114754992)

[1.1. Descripción general de la entidad 4](#_Toc114754993)

[1.2. Objetivo de estudio 5](#_Toc114754994)

[1.2.1. Flujo Actual de Procesos 5](#_Toc114754995)

[1.2.2. Análisis Actual de Procesos 6](#_Toc114754996)

[1.2.3. Procesos objeto de automatización 6](#_Toc114754997)

[1.3. Tendencias y Tecnologías Actuales 7](#_Toc114754998)

[1.3.1. Fundamentación de la Metodología utilizada 7](#_Toc114754999)

[1.4. Tendencias y tecnologías actuales 9](#_Toc114755000)

[1.4.1. Características y evaluación de los sistemas automatizados existentes, vinculados al campo de acción 9](#_Toc114755001)

[1.4.2. Fundamentación del Entorno de desarrollo, Lenguaje de programación, Gestor de Base de Datos, Tecnología en análisis 9](#_Toc114755002)

[1.5. Conclusiones Parciales 16](#_Toc114755003)

[**Capítulo II. Descripción de la propuesta de solución** 17](#_Toc114755004)

[[RESUMEN DEL CAPITULO] 17](#_Toc114755005)

[2.1. Ingeniería de requisitos 17](#_Toc114755006)

[2.1.1. Definición de requisitos funcionales 17](#_Toc114755007)

[2.1.2. Definición de requisitos no funcionales 21](#_Toc114755008)

[2.1.3. Actores del Sistema a automatizar 22](#_Toc114755009)

[2.1.4. Diagrama de Casos de Uso del Sistema 22](#_Toc114755010)

[2.1.5. Descripción de los casos de uso del Sistema 23](#_Toc114755011)

[2.2. Arquitectura del Sistema 32](#_Toc114755012)

[2.2.1. Fundamentación del estilo o patrón de arquitectura seleccionado 32](#_Toc114755013)

[2.3. Diseño e implementación del sistema 34](#_Toc114755014)

[2.3.1. Diagrama de clases de diseño (CU SIGNIFICATIVOS) 34](#_Toc114755015)

[2.3.2. Diagrama de secuencia (CU Significativos) 34](#_Toc114755016)

[2.3.3. Diseño de la base de datos 34](#_Toc114755017)

[2.3.4. Diagrama de componentes de despliegue 34](#_Toc114755018)

[[CONCLUSIONES] 34](#_Toc114755019)

# **Tabla de ilustraciones**

[Ilustración 1.1: Distribución geográfica de las tres sedes de la UCLV 4](#_Toc114044934)

[Ilustración 1 2: Evolución de RUP. (Fernández, 2000) 7](#_Toc114044935)

[Ilustración 1.3: Etiqueta básica de HTML con su apertura y cierre.(Coppola, 2022) 9](#_Toc114044936)

[Ilustración 1 4: Estructura y contenido + Presentación (CSS). (Baena, 2013) 10](#_Toc114044937)

[Ilustración 1.5: UML. (Fernández, 2000) 15](#_Toc114044938)

[Ilustración 1.6: Vistas y Diagramas UML (Rumbaugh et al., 1999) 16](#_Toc114044939)

# **Introducción**

A lo largo de la historia y en todo el mundo, el hombre ha tenido como uno de sus objetivos fundamentales obtener mas resultados con menos esfuerzos, utilizando técnicas que le permitan reducir el tiempo y mejorar la calidad de los procesos. Este ha sido el elemento detonador de un sin número de descubrimientos e inventos mecánicos y tecnológicos que constituyeron los antecedentes de la informatización.

En la Universidad Central “Marta Abreu” de las Villas, como el resto de las universidades del país, tienen entre sus metas la informatización y la transformación digital de los procesos sustantivos, lo cual se enmarca en el objetivo 7 del proyecto estratégico 2022 del MES que plantea: “Garantizar la transformación digital de las Universidades y Entidades de Ciencia, Tecnología e Innovación, en correspondencia con el perfeccionamiento de los procesos de la Educación Superior, y el posicionamiento efectivo en medios de comunicación y plataformas de Internet”.(*Proyecto Estratégico 2022*, 2022)

Algunos de los desarrollos aplicados en la informatización de nuestra Universidad han sido soluciones nacionales como el SIGENU para la gestión de los estudiantes, también se han desarrollado sistemas propios como el llamado SIGENU de posgrado para la gestión de las actividades de posgrados de la UCLV y en algunos casos se han ajustado softwares desarrollados en otras universidades a las condiciones de la UCLV como es el caso del sistema del control de la reserva para acceder el almuerzo en los comedores de trabajadores.

El área de la seguridad y protección de la Universidad es muy amplia y compleja por las propias características del centro. En particular el aspecto de la Guardia Obrero Estudiantil (GOE) es muy importante pues debe apoyar la formación de valores en estudiantes y trabajadores y también debe contribuir a la prevención de hechos delictivos en la Universidad y a su disminución. Estos hechos delictivos en la mayoría de los casos afectan directamente a los recursos destinados al funcionamiento del centro y por lo tanto afectan a estudiantes, trabajadores y cuadros en el cumplimiento de los objetivos propuestos. En la organización a nivel central de la GOE, cada Facultad o área universitaria es responsable de la planificación, realización y control de esta actividad en sus edificios, un grupo de serenos hacen guardia en áreas comunes de la Universidad y en la mayoría de la Facultades siempre que haya disponibilidad y los estudiantes a través de la cuartelaría apoyan la protección de las residencias estudiantiles, de esta manera debe quedar protegida en su totalidad la Universidad.

Actualmente no existe en la Facultad de Matemática, Física (FMFC) y Computación una herramienta informática que apoye GOE, por lo que, ese proceso y la información que de él pueda ser necesaria a todos los niveles se obtiene manualmente.

En la FMFC durante el primer periodo del curso 2022 han existido diferentes problemas con las actividades de planificación, ejecución y control de la GOE lo que ha incidido en el bajo porciento de asistencia a la GOE, sobre todo de estudiantes. Se ha identificado que los problemas en la planificación, la información tardía sobre la planificación y el no control de la asistencia a la GOE con las correspondientes medidas para los incumplidores son los factores que más han incidido sobre este proceso Por lo que desarrollar una herramienta de software que apoye la gestión del proceso de la GOE en la Facultad pudiera incidir en una planificación con una inmediatez en la información a través de la red UCLV, una mejor gestión en el control de la asistencia y las acciones que se derivan de ausentarse a la GOE y por ende incidir indirectamente en mejores resultados en la ejecución de esta actividad.

Teniendo en cuenta lo antes mencionado se define la siguiente **hipótesis** de investigación: con la implementación de un sistema para la GOE se logrará facilitar y mejorar la planificación, control y seguimiento de la GOE en la FMFC.

.

# **Problema Investigación**

¿Cómo contribuir a la informatización de la planificación, control y seguimiento de la GOE en la FMFC?

# **Objetivo General**

Desarrollar una aplicación web para la gestión de los procesos administrativos de planificación, control y seguimiento de la Guardia Obrero Estudiantil en la FMFC.

# **Objetivos específicos**

* Analizar toda la gestión que se realiza durante la planificación, ejecución y control de la Guardia Obrero Estudiantil en la FMFC.
* Diseñar y desarrollar una base de datos que facilite la gestión de los procesos administrativos de la GOE.
* Implementar una aplicación web que apoye las etapas de planificación, control y seguimiento de la Guardia Obrero Estudiantil en la FMFC.
* Validar la implementación de los módulos desarrollados en la aplicación web.

# **Preguntas**

* ¿Cuáles etapas del proceso actual de gestión de la Guardia Obrero Estudiantil en la FMFC pueden ser automatizados a través de una herramienta de software?
* ¿Cómo diseñar un modelo lógico de una base de datos que permita la captura de toda la información?
* ¿Qué gestor de base de datos seleccionar para facilitar los procesos administrativos en la FMFC?
* ¿Qué módulos deben ser implementados para desarrollar una aplicación web que apoye la gestión del proceso de GOE?

# **Estructura del documento**

(Todo el documento)

# **Capítulo I. Fundamentación teórica**

En este capítulo se exponen aspectos generales de la organización, planificación, control y seguimiento de la GOE en la FMFC. Asimismo, se exponen las tecnologías utilizadas para el desarrollo de la investigación y algunas ideas que fundamentan las decisiones tomadas.

## Descripción general de la entidad

La Universidad Central “Marta Abreu” de Las Villas, según se afirma en ("Portal Web Universidad Central "Marta Abreu" de Las Villas," 2022), “constituye el Centro de Educación Superior más importante de la región central de Cuba y es acreedora de la condición de Excelencia, otorgada por la Junta de Acreditación Nacional en 2016”.

Esta es la Universidad más interdisciplinaria del país, se estudian 54 carreras que abarcan las ciencias humanísticas, las técnicas y las naturales, organizadas en 12 facultades y, además, se ofrecen programas doctorales, programas académicos de Maestría y de especialidades.

Esta institución está integrada por tres sedes, ubicadas en áreas geográficas diferentes, ver Ilustración 1.1:

1. Sede Central.
2. Sede pedagógica “Félix Valera”.
3. Sede “Manuel Fajardo”.

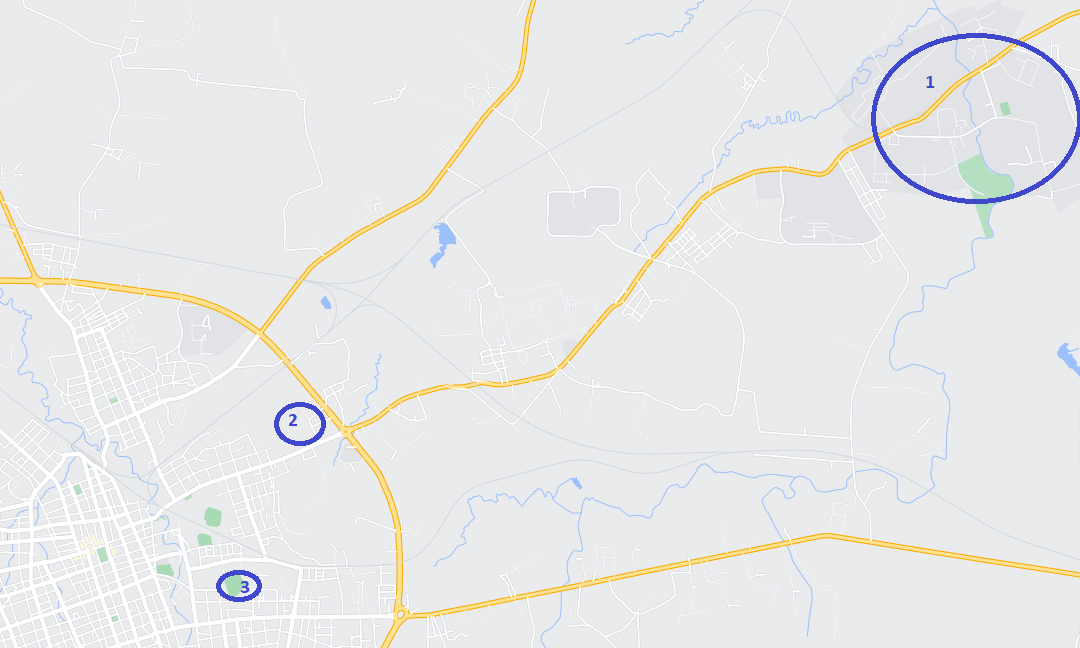


Ilustración 1.1.1.1: Distribución geográfica de las tres sedes de la UCLV

La Dirección de Defensa y Seguridad (DDS) de la UCLV se encarga de dar a conocer los planes especiales y el reglamento aprobado para cada facultad, así como la Resolución 176 de la Ley No. 1371 del 27 de noviembre de 1971, que se refiere a la aplicación de la política de seguridad y protección física en las instalaciones y demás bienes sociales asignados a los organismos y entidades estatales, así como los de las organizaciones sociales y de masas.

## Objetivo de estudio

“El objeto de estudio es la resultante de la intersección entre el problema que instala la necesidad de avanzar a través de un proceso de investigación y los conceptos, sus definiciones y relaciones que otorgan sentido al objeto y que hacen a su interpretación y comprensión”.(Néstor Cohen, 2019)

Elementos del diseño metodológico del objeto de estudio:

• Propósito.

• Enfoque.

• Dimensión temporal.

• Unidad de análisis.

• Recolección de datos.

• Tratamiento de datos.

### Flujo Actual de Procesos

Se tiene un Grupo de Atención a la GOE formado por el decano, el partido, el planificador de la GOE y la administración que establece la política de cómo se organizará la GOE.

Después de establecida la política, se le entrega el listado actual de estudiantes y trabajadores al planificador de la GOE (un profesor de Matemática) para que planifique la GOE.

Obtenida la planificación es revisada por el Grupo de Atención a la GOE y si no hay problemas se circula a los departamentos y secciones sindicales.

Una persona del departamento administrativo tiene asignado el proceso de control de la GOE que implica recordar la GOE a trabajadores y estudiantes y verificar la asistencia a la GOE. Cada vez que una persona hace GOE debe firmar un modelo que se encuentra en el local de la GOE y a través de ese modelo se controla asistencia. Esta persona que controla la GOE debe hacer un reporte semanal para de asistencia y el reporte mensual que se pide por PT Universitario.

El reporte semanal va al decano y de ahí a los PPA y a los profesores que atienden Cortes Disciplinarias. En las cortes se cita al estudiante, se le oye su argumentación, si es la primera vez se le reprograma la GOE y si ha incurrido más de una vez y se decide por los profesores se hace un proceso disciplinario como se establece. En la Facultad se ha organizado 3 grupos de cortes: Matemática-Física, Ciencia de la Información-Administración y Comp-CII con tres profesores y un estudiante en cada grupo de corte. Para los profesores que faltan a la GOE solo actúan los profesores, para los estudiantes dos profes y el estudiante del grupo de Corte.

El sistema de GOE universitario tiene 6 turnos:

T1: 7 am a 11 am

T2: 11 am a 3 pm

T3: 3 pm a 7 pm

T4: 7 pm a 11 pm

T5: 11 pm a 3 am

T6: 3 am a 7 am

Entre semana solo se cubren los turnos T4 al T6 y los fines de semana todos. Por la escases de profesores relativamente jóvenes en la facultad, los profesores hacen la GOE de 7 pm a 7 am (turnos T4 al T6), no así lo estudiantes entre semana y fines de semana que se les respeta sus 4 horas.

### Análisis Actual de Procesos

Partiendo del análisis del flujo actual del proceso se detectan los siguientes problemas:

1. Estas tareas se realizan utilizando herramientas como Word y Excel, que no brindan suficientes opciones para la planificación y gestión del proceso.

2. Algunos campos en el informe preparado pueden estar escritos incorrectamente.

3. No se garantiza la consistencia durante el almacenamiento de los datos.

4. No existe una herramienta efectiva para el análisis y toma de decisiones.

En vista de lo anterior, se decidió desarrollar un sistema para la gestión del GOE en la FMFC.

### Procesos objeto de automatización

Analizando el tedioso trabajo que realizan los encargados de la guardia, se decide automatizar los siguientes procesos para facilitar dicha tarea:

1. Planificación y control de la guardia obrera estudiantil.

2. Elaboración de los reportes de control de las guardias.

## Tendencias y Tecnologías Actuales

### Fundamentación de la Metodología utilizada

Proceso Unificado Racional (RUP)

El Proceso Unificado Racional (Rational Unified Process en inglés) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el Rational Method Composer (RMC), que permite la personalización de acuerdo a necesidades. (NGuerrero, 2018)

Objetivos de RUP (Fernández, 2000):

• Proporcionar una guía del orden de las actividades de los equipos.

• Especificar cuándo y cuáles artefactos deben ser desarrollados.

• Dirigir las tareas de desarrolladores individuales y equipos como una sola.

• Ofrecer criterios para monitorear y medir los productos y actividades del proyecto.

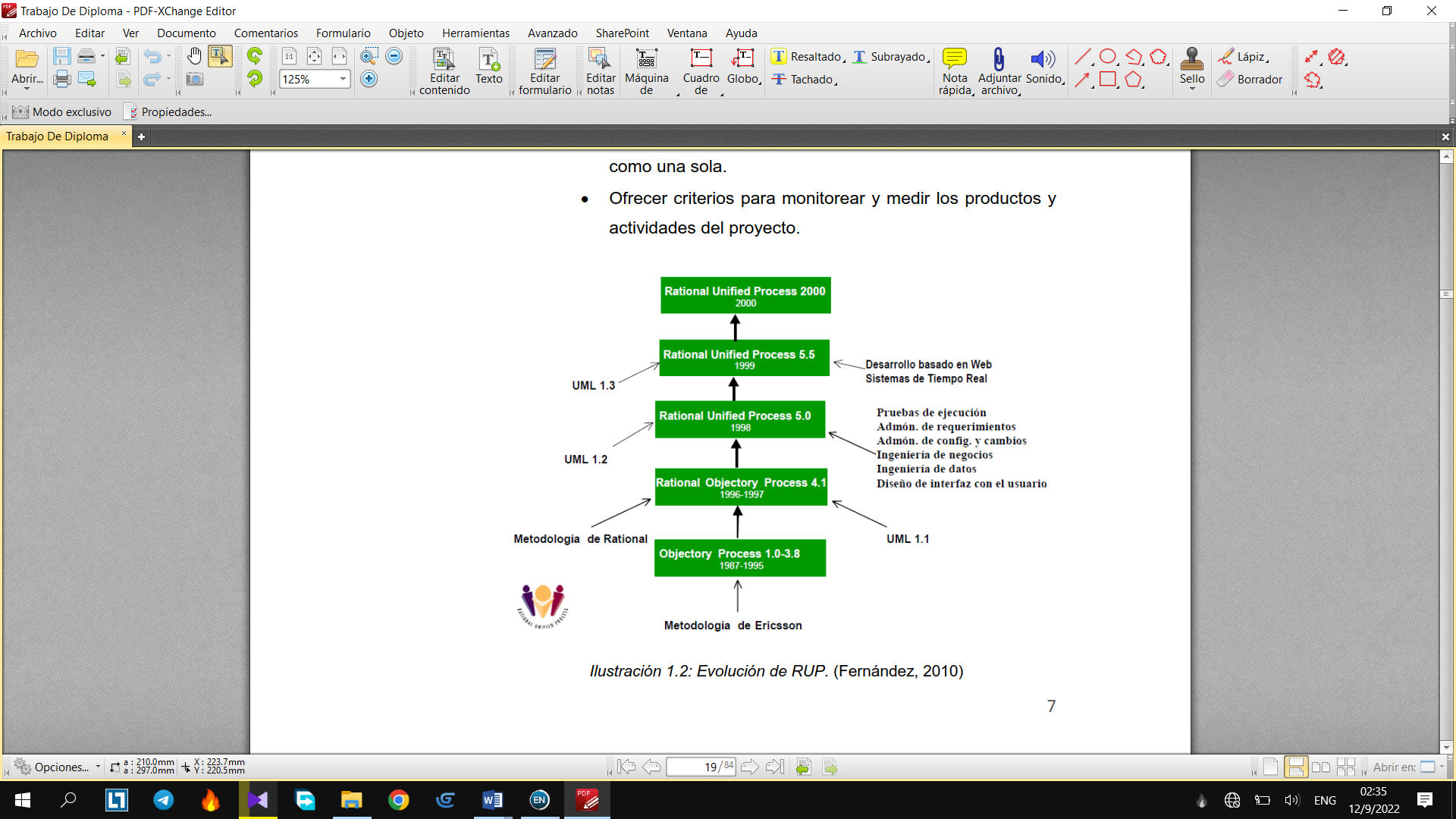


Ilustración 1 1.3.1: Evolución de RUP. (Fernández, 2000)

La estructura dinámica del RUP permite que el proceso de desarrollo se fundamentalmente iterativo en las cuatro fases.

Fases (Proyecto\_Grado\_UNAD, 2014):

1. Fase de inicio: (se define el alcance del proyecto)

Se define el alcance del proyecto con los clientes, se identifican los riesgos asociados al proyecto, se elabora el plan de las fases y el de la iteración posterior, se detalla de manera general la arquitectura del software.

1. Fase de Elaboración: (Definición, análisis y diseño)

Se diseña la solución preliminar, se selecciona los casos de uso que permiten definir la arquitectura base del sistema y se desarrollara el primer análisis del dominio del problema.

1. Fase de desarrollo o construcción: (implementación)

La función de esta fase es completar la funcionalidad del sistema, se clarifican los requisitos pendientes, se administran los cambios de acuerdo a las evaluaciones realizadas por los usuarios, y se realizan las mejoras para el proyecto.

1. Fase de transición: (fin del proyecto y pruebas)

Fase de cierre, el propósito es asegurar que le software esté disponible para los usuarios finales, se ajustan los errores y defectos encontrados en las pruebas de aceptación, se capacitan a los usuarios y se provee el soporte necesario.

Esta metodología comprende 3 principios claves:

1. Dirigido por casos de uso: La razón de ser de un sistema es servir a los usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el sistema debe proveer a sus usuarios. Estos constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo, incluyendo el diseño, la implementación y las pruebas del sistema. Los casos de uso representan los requisitos funcionales del sistema.

2. Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas de software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requisitos no funcionales.

3. Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos, para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

## Tendencias y tecnologías actuales

### Características y evaluación de los sistemas automatizados existentes, vinculados al campo de acción

### Fundamentación del Entorno de desarrollo, Lenguaje de programación, Gestor de Base de Datos, Tecnología en análisis

#### HTML

Siglas en inglés de HyperText Markup Language (lenguaje demarcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web.

HTML es un lenguaje que se utiliza para la creación de páginas web. Por página entendemos el documento que aparece en el visualizador o navegador. (Murcia, 2011)

Es importante recalcar que el lenguaje HTML no es un lenguaje de programación, sino de marcado. Esto significa que solo sirve para crear páginas web estáticas. Sin embargo, sí es posible combinar este lenguaje con otros para desarrollar sitios mucho más dinámicos y atractivos.

La estructura HTML de una página web se compone de las etiquetas <html>, que indica el comienzo de la página; <head>, que contiene información más descriptiva sobre el sitio; y <body>, que conforma todos los elementos visibles de la página.

Todas las etiquetas están constituidas al inicio y al final por los signos < y >, que representan precisamente la apertura y el cierre de cada elemento para que el lenguaje HTML pueda funcionar. En este sentido, existen dos tipos de etiquetas HTML:

* Etiquetas emparejadas. Vienen en pares y tienen una apertura y cierre. Tal como las que conoces: < >.
* Etiquetas vacías. Se refiere a las etiquetas que no requieren cierre.



Ilustración 1.1.4.1: Etiqueta básica de HTML con su apertura y cierre.(Coppola, 2022)

Los atributos se refieren a toda la propiedad adicional de las etiquetas HTML. Esto significa que definen cualquier elemento de la página web; por ejemplo, ancho, entrada, texto. Estos también ayudan a almacenar información en metaetiquetas con la atribución de nombre, contenido y tipo.

#### CSS

CSS (Cascade Style Sheets) es un estándar del W3C que define las reglas para la presentación del contenido de un documento html o xhtml. (Baena, 2013)

Permite definir el aspecto de un elemento o un grupo de elementos dentro de un documento en un medio determinado.

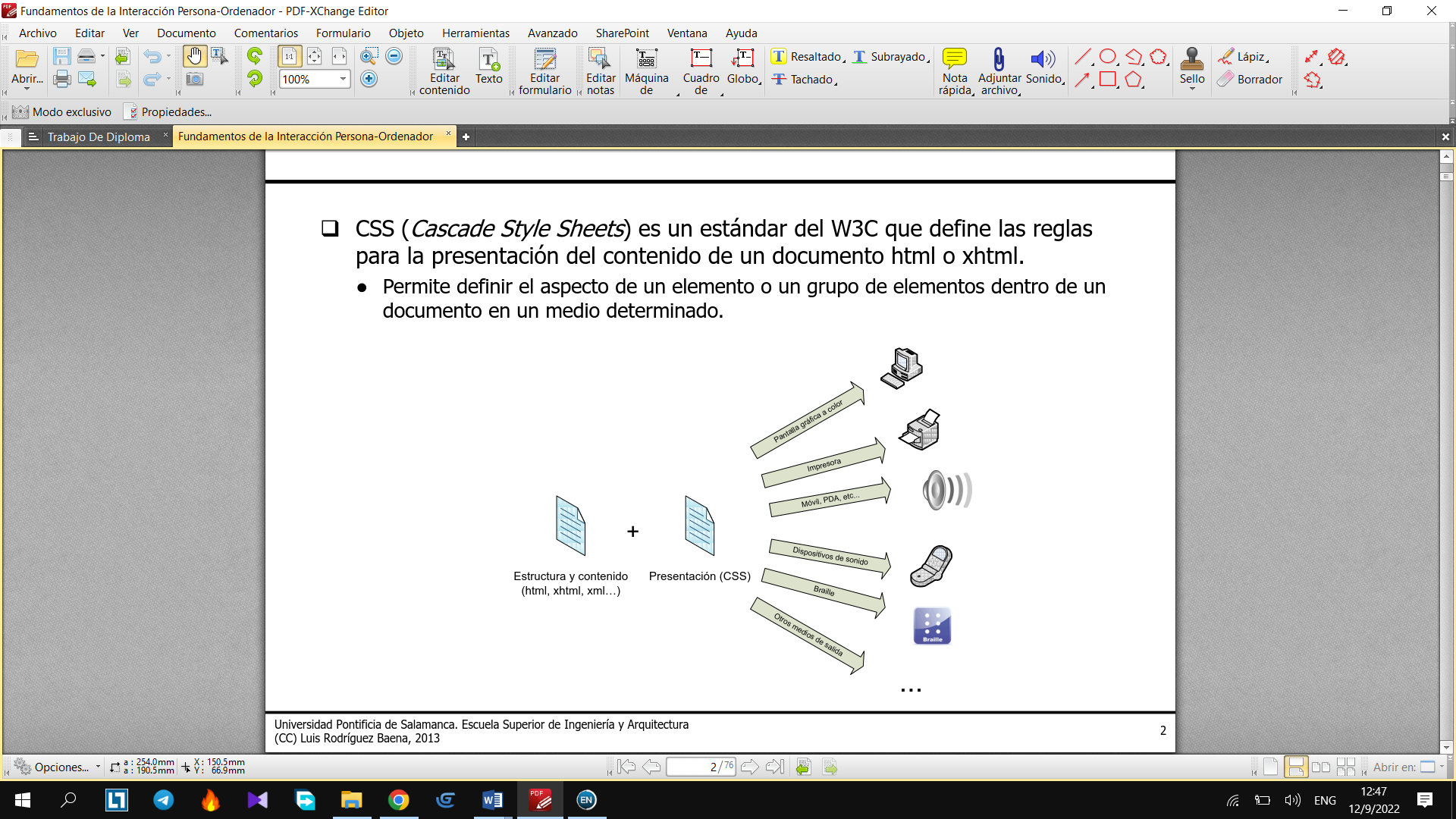


Ilustración 1 1.4.2: Estructura y contenido + Presentación (CSS). (Baena, 2013)

El lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc.

Existen tres opciones para incluir CSS en un documento HTML.

1. Incluir CSS en el mismo documento HTML
2. Definir CSS en un archivo externo
3. Incluir CSS en los elementos HTML

CSS presenta varias ventajas y desventajas

Ventajas (Baena, 2013):

* Simplifica el código de las páginas web.
* Optimiza el rendimiento de los navegadores.
* Permite dar una apariencia homogénea a un sitio web al aplicar los mismos estilos a todas sus páginas.
* Amplía las posibilidades de presentación de HTML al permitir mucho más control.
* Permite presentar el mismo contenido de diferentes formas sin tener que modificarlo.
* Permite que los usuarios con necesidades especiales creen sus propias hojas de estilo para ver el contenido según sus preferencias.

Desventajas (Baena, 2013):

* No todos los navegadores soportan por igual todas las características de CSS.
* Se debe procurar que el contenido de la página se pueda mostrar correctamente y en su totalidad sin aplicar las horas de estilo.

#### Python

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario “traducirlo” a lenguaje máquina.

Python es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. Con el paso del tiempo, Python ha ido ganando adeptos gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge.

Python tiene diversos usos (Cristancho, 2022):

* Análisis de Big Data: cuando se trata de analizar datos, y que a su vez pueden presentarse en grandes cantidades, Python es excelente. Cuenta con todo tipo de bibliotecas de procesamiento que le permiten analizar una gran cantidad de datos y en tiempo real.
* Data science: gracias a su uso de análisis de Big Data, este también es utilizado por científicos a la hora de estudiar una gran cantidad de datos imposible de hacer por cuenta propia.
* Inteligencia Artificial: Python es bien conocido por ser un lenguaje de programación que permite el desarrollo de Inteligencia Artificial de gran eficacia. Permitiendo así, el avance de esta tecnología que hace de nuestra vida algo más sencillo.
* Videojuegos: este lenguaje cuenta con herramientas que permiten la manipulación y creación de gráficos 3D. Esto permite trabajar con gráficos o desarrollar videojuegos interactivos.
* Machine learning: otro de los usos más comunes del Python, o donde resalta su potencial, es en el Machine Learning. Permite otorgar no solo inteligencia a las máquinas o robots, sino que los dota de capacidad de aprender por medio de la experiencia, pudiendo realizar nuevas tareas con el paso del tiempo.
* Desarrollo web: el desarrollo de aplicaciones o sitios web es también uno de los fuertes de Python, pudiendo crearlas con un sistema estable y eficiente en cuanto a funcionalidades.
  + - 1. *Django*

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Django ayuda a escribir software que es :

* Completo

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran que tenga "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

* Versátil

Django puede ser usado para construir casi cualquier tipo de sitio web desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc). ¡El sitio que estás leyendo actualmente está basado en Django!

Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos, motores de plantillas, etc.), también puede ser extendido para usar otros componentes si es necesario.

* Seguro

Django ayuda a los desarrolladores evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para "hacer lo correcto" para proteger el sitio web automáticamente. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable (en lugar de eso las cookies solo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un hash de contraseñas.

* Escalable

Django usa un componente basado en la arquitectura “shared-nothing” (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Teniendo en cuenta una clara separación entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores de cache, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas (por ejemplo, Instagram y Disqus, por nombrar solo dos).

* Mantenible

El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "Don't Repeat Yourself" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Django también promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos (siguiendo el patrón Model View Controller (MVC)).

* Portable

Django está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django.

#### PostgreSQL

PostgreSQL es un gestor de bases de datos relacional y orientado a objetos. Su licencia y desarrollo es de código abierto, siendo mantenida por una comunidad de desarrolladores, colaboradores y organizaciones comerciales de forma libre y desinteresadamente.

Características

* Es de código abierto: una de las principales razones por la cual PostgreSQL se ha vuelto tan popular es que se trata de un sistema de código abierto. Esto ha permitido que una gran comunidad de desarrolladores crezca para respaldarlo y continuar mejorándolo. Gracias a todo el apoyo con el que cuenta ha logrado transformarse en uno de los mejores gestores de bases de datos a nivel mundial.
* Es gratuito: como cabe esperarse se trata de un sistema totalmente gratis, no tenemos que pagar nada por utilizarlo. Cualquier persona es libre de descargar PostgreSQL desde su sitio web oficial y darle uso sin ningún costo.
* Es multiplataforma: una característica genial que de hecho es común en muchos grandes proyectos de código abierto es el hecho de que se trata de software multiplataforma, es decir, es un software que puede correr bajo distintos entornos y sistemas operativos, y es compatible con muchos de los servidores web más populares como Apache, Nginx y LiteSpeed por mencionar algunos
* Es fácil de usar: la facilidad de uso de PostgreSQL es sin dudas otra de las principales características de este sistema. Su administración se vuelve muy sencilla por medio de paneles con PgAdmin, que básicamente viene a ser un phpMyAdmin orientado para PostgreSQL. La posibilidad de realizar diversos procedimientos en forma sencilla hacen que PgAdmin sea ampliamente utilizado, aunque también permite realizar tareas más complejos, así que tanto novatos como usuarios expertos hacen uso de él.
* Puede manejar un gran volumen de datos: una característica extremadamente importante de PostgreSQL es su gran capacidad para el manejo de grandes volúmenes de datos, algo en lo que otros sistemas como MySQL aún no hacen tan bien. Las bases de datos de gran tamaño pueden hacer pleno uso del MVCC de PostgreSQL, resultando en un gran rendimiento. MVCC es un método de control que nos permite realizar tareas de escritura y lectura simultáneamente.
* Soporte total de ACID: otro punto muy importante que no se debe dejar de lado es el cumplimiento de ACID. ¿Qué es ACID? Estas siglas en inglés refieren a: atomicity, consistency, isolation y durability, que si lo traducimos al español básicamente hablan de la atomicidad, consistencia, aislamiento y durabilidad de las transacciones que se realizan en una base de datos. ¿Y por qué es tan importante? Porque tener soporte completo de ACID da la seguridad de que, si se produce una falla durante una transacción, los datos no se perderán ni terminarán donde no deban.

#### Unified Modeling Language (UML)

El Lenguaje Unificado de Modelado (Unified Modeling Language, UML) es un lenguaje de modelado visual que se usa para especificar, construir y documentar artefactos de un sistema. Captura decisiones y conocimientos sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. (Rumbaugh, Jaconson, & Booch, 1999)

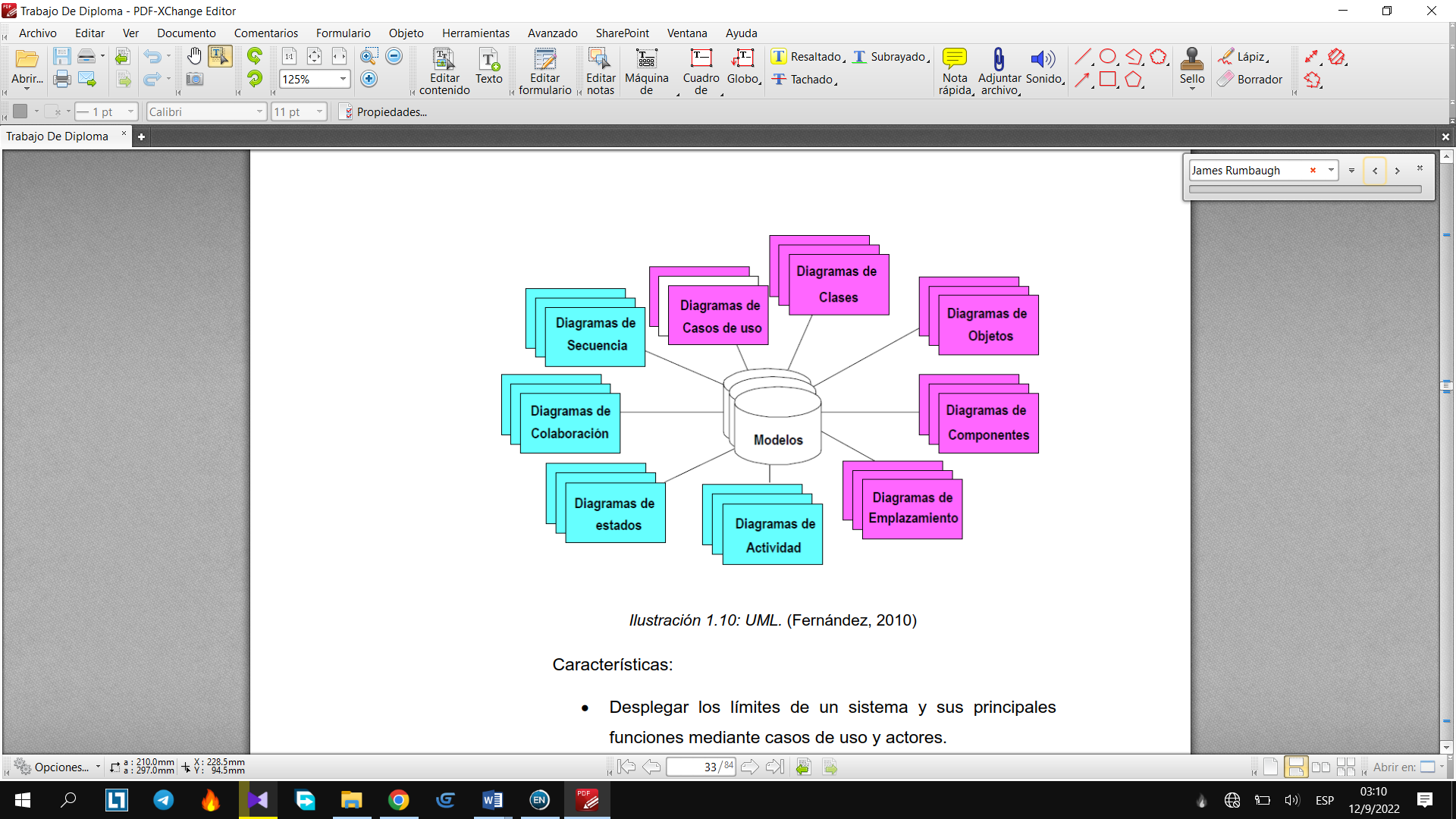


Ilustración 1.1.4.3: UML. (Fernández, 2000)

Características:

* Desplegar los límites de un sistema y sus principales funciones mediante casos de uso y actores.
* Representar la estructura estática de un sistema usando diagramas de clases.
* Modelar los límites de un objeto con diagramas de estados.
* Mostrar la arquitectura de la implementación física con diagramas de componentes y de emplazamiento o despliegue.

Conceptos y modelos de UML:

* Estructura estática.
* Comportamiento dinámico.
* Construcciones de implementación.
* Organización del modelo.
* Mecanismos de extensión.

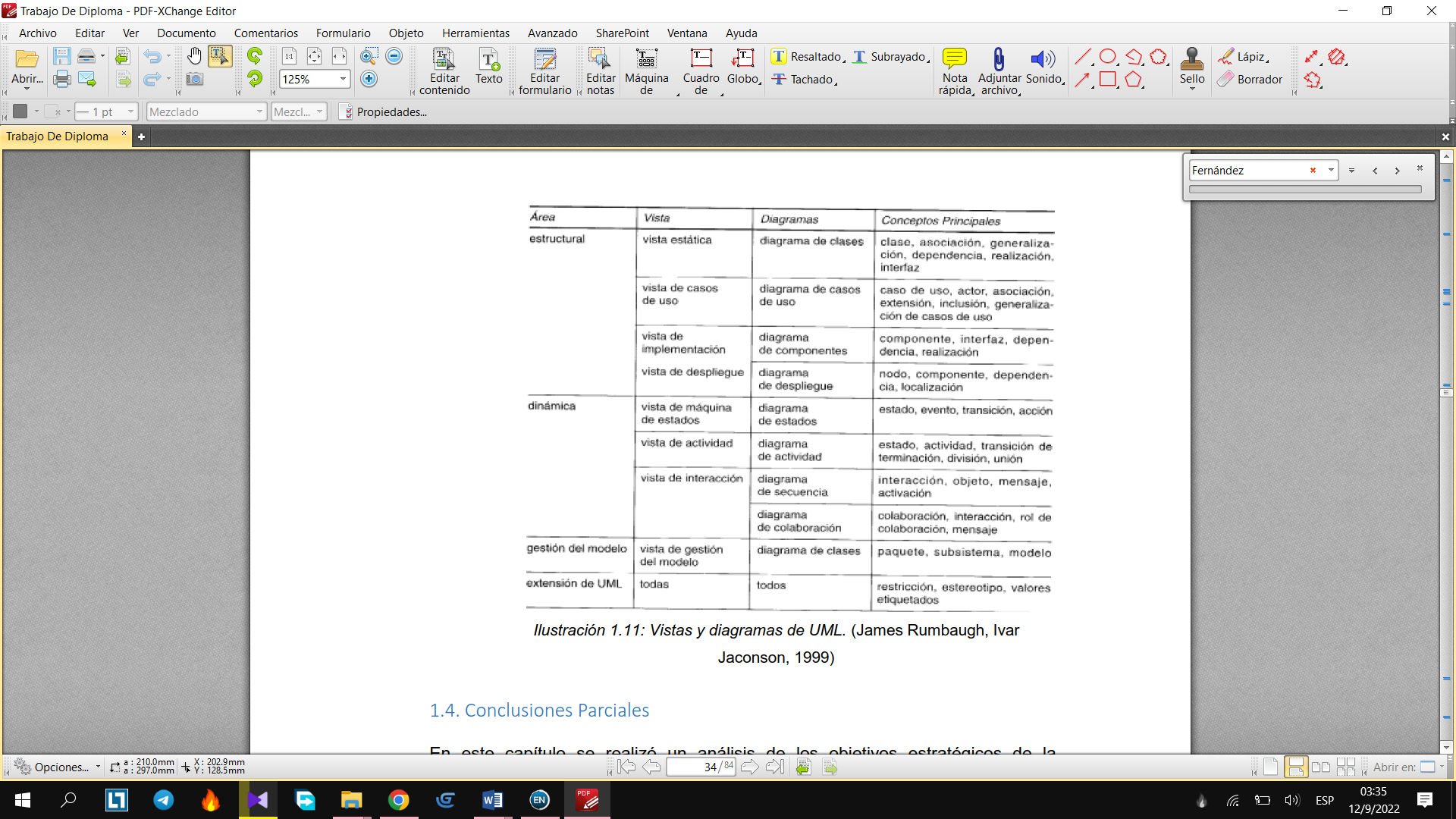


Ilustración 1.1.4.4: Vistas y Diagramas UML (Rumbaugh et al., 1999)

## Conclusiones Parciales

En este capítulo se realizó un análisis de los objetivos estratégicos de la Universidad Central “Marta Abreu” de Las Villas y con ello su misión de hacer más efectiva la guardia obrera estudiantil, más en concreto en la Facultad de Matemática, Física y Computación. Se realizó un estudio de la metodología y tecnologías a emplear para dar solución al problema descrito, decidiéndose utilizar: Pyhton como lenguaje de programación base junto con el framework Django; RUP como metodología de desarrollo de software; UML como lenguaje de modelado y PostgreSQL como gestor de base de datos para la herramienta.

# **Capítulo II. Descripción de la propuesta de solución**

## [RESUMEN DEL CAPITULO]

## 2.1. Ingeniería de requisitos

### 2.1.1. Definición de requisitos funcionales

“Los requerimientos funcionales son las descripciones explicitas del comportamiento que debe tener una solución de software y que información debe manejar.”(Analysis, 2015)

Requisitos funcionales que presenta el sistema:

1. Gestionar Listado de Estudiantes.

El sistema debe listar a cada alumno que integra la matrícula docente, con la información de cada uno de ellos.

* 1. Cargar estudiantes.

El sistema debe ser capaz de mostrar al usuario el listado de los estudiantes de la facultad.

* 1. Añadir un nuevo estudiante.

El sistema debe permitir al usuario agregar un nuevo estudiante con los datos apropiados.

* 1. Modificar datos de un estudiante existente.

El sistema debe brindar la posibilidad de actualizar los datos de un estudiante.

* 1. Eliminar un estudiante.

El sistema debe brindar la posibilidad de eliminar un estudiante con todos los datos del mismo.

1. Gestionar Listado de Trabajadores.

El sistema debe listar a todos los empleados que integran la plantilla de la facultad, junto con la información de cada uno de ellos.

* 1. Cargar trabajadores.

El sistema debe ser capaz de mostrar al usuario el listado de los trabajadores.

* 1. Añadir un nuevo trabajador

El sistema debe permitir al usuario agregar un nuevo trabajador con los datos apropiados.

* 1. Modificar datos de un trabajador existente.

El sistema debe brindar la posibilidad de actualizar los datos de un trabajador.

* 1. Eliminar un trabajador

El sistema debe brindar la posibilidad de eliminar un trabajador con todos los datos del mismo.

1. Gestionar Planificaciones.

El sistema debe ser capaz de listar todas las planificaciones de estudiantes y trabajadores.

* 1. Añadir una nueva planificación.

El sistema debe permitir añadir una nueva planificación con los datos que el usuario introduzca.

* 1. Modificar una planificación existente.

El sistema debe brindar la posibilidad de actualizar los datos de una planificación.

* 1. Eliminar una planificación.

El sistema debe brindar la posibilidad de eliminar una planificación con todos los datos de la misma.

1. Gestionar Excepciones.

El sistema debe brindar la posibilidad de listar todas las excepciones en una planificación tanto de estudiantes como trabajadores.

* 1. Añadir una excepción.

El sistema debe permitir al usuario insertar una nueva persona exceptuada con los correspondientes datos de la misma.

* 1. Modificar una excepción.

El sistema debe brindar la posibilidad de actualizar los datos de una persona exenta existente.

* 1. Eliminar una excepción.

El sistema debe brindar la posibilidad de eliminar una persona exenta existente, con todos los datos de la misma.

1. Gestionar Ausencias.

El sistema debe poder brindar al usuario la posibilidad de poder listar todas las ausencias que presentan los estudiantes y trabajadores.

* 1. Añadir una nueva ausencia.

El sistema debe permitir añadir una nueva ausencia con los datos que el usuario introduzca.

* 1. Modificar una ausencia existente.

El sistema debe brindar la posibilidad de actualizar los datos de una ausencia.

* 1. Eliminar una ausencia.

El sistema debe brindar la posibilidad de eliminar una ausencia con todos los datos de la misma.

1. Gestionar Causas de las Ausencias.

El sistema debe permitir al usuario listar todas las causas posibles de las ausencias.

* 1. Adicionar una causa

El sistema debe permitir al usuario adicionar una causa.

* 1. Modificar una causa.

El sistema debe brindar la posibilidad de actualizar una causa.

* 1. Eliminar una causa.

El sistema debe brindar la posibilidad de eliminar una causa existente.

1. Gestionar Días Feriados.

El sistema debe permitir al usuario listar las fechas que son feriados.

* 1. Añadir un día feriado.

El sistema debe permitir añadir un día feriado con los datos que el usuario introduzca.

* 1. Eliminar un día feriado.

El sistema debe brindar la posibilidad de eliminar un día feriado.

1. Gestionar Reportes Mensuales.

El sistema debe listar todos los reportes realizados, junto con la información de cada uno de ellos.

* 1. Crear reporte.

El sistema debe permitir crear un reporte con los datos que el usuario introduzca.

* 1. Generar reporte.

El sistema debe brindar la posibilidad de generar un reporte con la información pertinente que debe conformar el mismo.

* 1. Eliminar reporte.

El sistema debe brindar la posibilidad de eliminar un reporte existente.

1. Gestionar Cortes.

El sistema debe permitir al usuario listar todas las cortes que se han realizado hasta el momento con la información de los mismos.

* 1. Añadir una corte.

El sistema debe permitir añadir una corte a un estudiante con sus respectivos datos.

* 1. Modificar una corte.

El sistema debe brindar la posibilidad de actualizar una corte realizada a un estudiante, así como todos sus datos.

* 1. Eliminar una corte

El sistema debe brindar la posibilidad de eliminar una corte con todos los datos de la misma.

1. Gestionar el Reglamento.

El sistema debe de ser capaz de que todos los usuarios tengan acceso al reglamento de la facultad y descargarla.

* 1. Subir el reglamento.

El sistema debe brindar la posibilidad de subir un reglamento al sitio.

* 1. Eliminar un reglamento existente.

El sistema debe brindar la posibilidad de eliminar un reglamento existente.

1. Imprimir GOE

El sitio debe de ser capaz de generar todos los documentos de los diferentes tipos de guardias para su impresión.

1. Gestionar Facultad.

El sistema debe permitir añadir, modificar y eliminar usuarios con diferentes roles.

### 2.1.2. Definición de requisitos no funcionales

Los requisitos no funcionales especifican propiedades del sistema, como pueden ser la seguridad y la fiabilidad del mismo. Estos requisitos no solo hacen referencia al sistema que se va a desarrollar, también consideran algunas restricciones, es decir, se encargan de definir aspectos como, por ejemplo, que estándares de calidad se deben seguir en el desarrollo del sistema. (Guerrero & Delgado, 2010)

Dentro de los mismos encontramos:

* Requisitos del Software

1. (QUE SERVIDOR WEB DEBO USAR)
2. El sistema trabaja con el gestor de base de datos PostgreSQL.
3. Las PC clientes deben tener instalado un Navegador Web como Chrome, Mozilla Firefox, Internet Explorer u Opera.

* Rendimiento

1. El sistema debe estar operativo las 24 horas del día.
2. El sistema debe interactuar con la Base de Datos lo más rápido posible para satisfacer las solicitudes de los usuarios
3. El sistema debe poder manejar todos los datos que se suban o descarguen del sitio.
4. El sistema debe ser capaz de soportar un amplio número de usuarios conectados a la vez.

* Soporte

1. El mantenimiento que necesite el sitio debe ser sencillo y factible.

* Requisitos de portabilidad

1. El sistema no requiere de una instalación previa debido a que está desarrollado en una plataforma web.
2. El sistema puede ejecutarse en cualquier sistema operativo Windows 95 o superior, así como distribuciones Linux.

* Usabilidad

1. El sistema será de fácil acceso y uso para cualquier usuario que acceda a él, necesitando un mínimo de conocimientos básicos de computación.

* Requisito de interfaz

1. El sistema presenta una interfaz sencilla, amigable y fácil de usar de manera que facilite el trabajo con el mismo.

* Seguridad

1. La seguridad del sitio esta implementada a través de un sistema de roles de usuario, donde cada uno de ellos tendrá una función definida y acceso a ejecutar solo las acciones que le correspondan.
2. La Base de datos estará protegida contra intrusos externos y solo los administradores podrán modificar el contenido de la misma.

* Ayuda y documentación

1. La aplicación brinda la ayuda al usuario para comprender fácilmente como se debe usar la misma.

### 2.1.3. Actores del Sistema a automatizar

El actor es una entidad externa que tiene interés en interactuar con el sistema. A menudo, es una persona que usa el sistema, pero también puede ser otro sistema o alguna clase de dispositivo hardware que necesita interactuar con el sistema. (Mediavilla)

Actores del sistema:

Responsable de la planificación y control de la GOE:

Es quien atiende directamente el proceso de planificación y control de la guardia.

Visualiza y actualiza el listado de estudiantes y trabajadores, imprime los diferentes tipos de

guardias, gestiona reportes mensuales, gestiona los cortes y gestiona todo acerca del reglamento

de la facultad.

* Administrador.

Gestiona la facultad, además de las fechas de días feriados.

* Trabajador.

Es quien visualiza el listado de las guardias asignadas durante el proceso de planificación y el control de las mismas.

* Estudiante.

Es quien visualiza el listado de las guardias asignadas durante el proceso de planificación y el control de las mismas.

### 2.1.4. Diagrama de Casos de Uso del Sistema

### 2.1.5. Descripción de los casos de uso del Sistema

## 

|  |  |
| --- | --- |
| **Caso de uso del sistema** | **Gestionar Estudiantes** |
| Actor | Responsable de la planificación y control de la GOE. |
| Propósito | Gestionar toda la información referente a los Estudiantes. |
| Resumen | Inicia cuando el Responsable de la planificación y control de la GOE selecciona la opción “Lista de Estudiantes”, seguidamente puede insertar, actualizar o eliminar un estudiante y finaliza cuando el Responsable de la planificación y control de la GOE termina de ejecutar una de estas tres operaciones. |
| Responsabilidades | Gestionar la información relacionada a los Estudiantes. |
| Casos de uso asociados |  |
| Precondiciones | Conexión a la base de datos |

**PROTOTIPO DE INTERFAZ**

|  |  |
| --- | --- |
| **Flujo normal de eventos. Sección A: Insertar Estudiante** | |
| Acción del actor | Respuesta del sistema |
| 1. El Responsable de la planificación y control de la GOE selecciona la opción de “Lista de Estudiantes”. |  |

|  |  |
| --- | --- |
| 3. El Responsable de la planificación y control de la GOE selecciona la opción añadir.    5. El Responsable de la planificación y control de la GOE llena los campos y acepta. | 2. El sistema muestra una tabla con los datos referentes a cada una de los estudiantes.        4. El sistema muestra un formulario con los datos necesarios para añadir un estudiante.         1. El sistema introduce los datos en la base de datos. 2. Muestra un mensaje de éxito. |
| **Flujos alternativos 1** | |
| 5. El Responsable de la planificación y control de la GOE llena los campos y acepta.    7. El Responsable de la planificación y control de la GOE verifica todos los campos y acepta. | 6. El sistema muestra un mensaje de error donde indica que existen campos vacíos. |
| **Flujos alternativos 2** | |
| 5. El Responsable de la planificación y control de la GOE llena los campos y acepta.        7. El Responsable de la planificación y control de la GOE verifica todos los campos y acepta. | 6. El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |
| **Flujo normal de los eventos. Sección B: Modificar Estudiante** | |
| Acción del actor | Respuesta del sistema |

|  |  |
| --- | --- |
| 1. El Responsable de la planificación y control de la GOE selecciona un estudiante de los existentes. 2. El Responsable de la planificación y control de la GOE hace click en el ícono “Editar”.       4. El Responsable de la planificación y control de la GOE edita los campos del estudiante que desee y acepta. | 3. El sistema carga el estudiante correspondiente y muestra sus datos en un formulario.         1. El sistema actualiza los datos del estudiante en la base de datos. 2. Muestra mensaje de éxito. |
| **Flujos alternativos 1** | |
| 4. El Responsable de la planificación y control de la GOE edita los campos del estudiante que desee y acepta.      6. El Responsable de la planificación y control de la GOE verifica que los campos estén llenos y acepta. | 5. El sistema muestra un mensaje de error donde indica que algún campo ha quedado vacío. |
| **Flujos alternativos 2** | |
| 4. El Responsable de la planificación y control de la GOE edita los campos del estudiante que desee y acepta.        6. El Responsable de la planificación y control de la GOE verifica que los campos estén correctos y acepta. | 5. El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |
| **Flujo normal de los eventos. Sección C: Eliminar Estudiante** | |
| Acción del actor | Respuesta del sistema |
| 1. El Responsable de la planificación y control de la GOE selecciona un estudiante. 2. El Responsable de la planificación y control de la GOE hace click en el ícono “Eliminar”.     4. El Responsable de la planificación y control de la GOE hace click en el botón aceptar. | 3. El sistema muestra un cartel de confirmación.       1. El sistema elimina los datos del estudiante en la base de datos. 2. Muestra mensaje de éxito. |
| Post condiciones | La base de datos debe quedar actualizada de acuerdo a las operaciones realizadas en las secciones A, B y C. |

|  |  |  |
| --- | --- | --- |
| **Caso de uso del sistema** | **Gestionar Planificación** | |
| Actor | Responsable de la planificación y control de la GOE. | |
| Propósito | Gestionar toda la información referente a la gestión y control de las planificaciones. | |
| Resumen | Inicia cuando el Responsable de la planificación y control de la GOE selecciona alguna de las diferentes “Planificaciones”, seguidamente puede insertar, actualizar o eliminar una planificación y finaliza cuando el Responsable de la planificación y control de la GOE termina de ejecutar una de estas tres operaciones. | |
| Responsabilidades | Gestionar la información relacionada con las planificaciones de los estudiantes y trabajadores. | |
| Casos de uso asociados |  | |
| Precondiciones | Conexión a la base de datos | |
| **PROTOTIPO DE INTERFAZ** | |
| **Flujo normal de eventos. Sección A: Generar Planificación** | | |
| Acción del actor | Respuesta del sistema | |
| 1. El Responsable de la planificación y control de la GOE selecciona la opción de “Planificación”.        3. El Responsable de la planificación y control de la GOE selecciona la opción generar.      5. El Responsable de la planificación y control de la GOE escoge la forma de planificación que necesite. | 2. El sistema muestra una tabla con los datos referentes a cada una de las planificaciones de los estudiantes o trabajadores.        4. El sistema muestra una interfaz con las formas de planificar la guardia.         1. El sistema realiza la planificación e inserta los datos en la base de datos. 2. Muestra un mensaje de éxito. | |

|  |  |
| --- | --- |
| **Flujo normal de eventos. Sección B: Nueva Planificación** | |
| Acción del actor | Respuesta del sistema |
| 1. El Responsable de la planificación y control de la GOE selecciona una de las diferentes “Planificaciones”.      3. El Responsable de la planificación y control de la GOE selecciona la opción insertar.      5. El Responsable de la planificación y control de la GOE llena los campos y acepta. | 2. El sistema muestra una tabla con los datos de la planificación seleccionada referente a estudiantes o trabajadores.      4. El sistema muestra un formulario con los datos necesarios para insertar una planificación.       1. El sistema introduce los datos en la base de datos. 2. Muestra un mensaje de éxito. |
| **Flujos alternativos 1** | |
| 5. El Responsable de la planificación y control de la GOE llena los campos y acepta.      7. El Responsable de la planificación y control de la GOE verifica todos los campos y acepta. | 6. El sistema muestra un mensaje de error donde indica que existen campos vacíos. |
| **Flujos alternativos 2** | |
| 5. El Responsable de la planificación y control de la GOE llena los campos y acepta. | 6. El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |

|  |  |
| --- | --- |
| 7. El Responsable de la planificación y control de la GOE verifica todos los campos y acepta. |  |
| **Flujo normal de los eventos. Sección C: Actualizar Planificación** | |
| Acción del actor | Respuesta del sistema |
| 1. El Responsable de la planificación y control de la GOE selecciona una planificación para modificar.   3. El Responsable de la planificación y control de la GOE edita los campos de la planificación que desee y acepta. | 2. El sistema carga la planificación correspondiente y muestra sus datos en un formulario.       1. El sistema actualiza los datos de la planificación en la base de datos. 2. Muestra mensaje de éxito. |
| **Flujos alternativos 1** | |
| 3. El Responsable de la planificación y control de la GOE edita los campos de la planificación que desee y acepta.      5. El Responsable de la planificación y control de la GOE verifica que los campos estén llenos y acepta. | 4. El sistema muestra un mensaje de error donde indica que algún campo ha quedado vacío. |
| **Flujos alternativos 2** | |
| 3. El Responsable de la planificación y control de la GOE edita los campos de la planificación que desee y acepta. | 4. El sistema muestra un mensaje de error donde indica que algún campo contiene caracteres incorrectos. |
| 5. El Responsable de la planificación y control de la GOE verifica que los campos estén correctos y acepta. |  |
| **Flujo normal de los eventos. Sección D: Eliminar Planificación** | |
| Acción del actor | Respuesta del sistema |
| 1. El Responsable de la planificación y control de la GOE selecciona una de las diferentes planificaciones. 2. El Responsable de la planificación y control de la GOE hace click en el ícono “Eliminar”.       4. El Responsable de la planificación y control de la GOE hace click en el botón aceptar. | 3. El sistema muestra un cartel de confirmación.         1. El sistema elimina la planificación en la base de datos. 2. Muestra mensaje de éxito. |
| Post condiciones | La base de datos debe quedar actualizada de acuerdo a las operaciones realizadas en las secciones A, B, C y D. |

## 2.2. Arquitectura del Sistema

La arquitectura utilizada para la implementación de la aplicación es Modelo Vista Controlador (MVC)

### 2.2.1. Fundamentación del estilo o patrón de arquitectura seleccionado

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. (Alicante)

* El Modelo que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
* La Vista, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
* El Controlador, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El modelo es el responsable de:

* Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
* Define las reglas de negocio.
* Lleva un registro de las vistas y controladores del sistema.

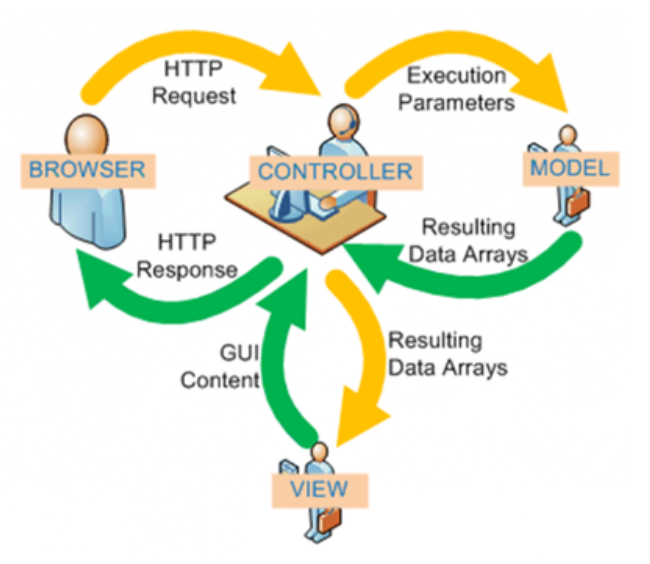
El controlador es responsable de:

* Recibe los eventos de entrada (un clic, un cambio en un campo de texto).
* Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas.

Las vistas son responsables de:

* Recibir datos del modelo y mostrarlos al usuario.
* Tienen un registro de su controlador asociado.
* Pueden dar el servicio de "Actualización()", para que sea invocado por el controlador o por el modelo.

El flujo que sigue el control generalmente es el siguiente:



ENUMERAR ILUSTRACION (Alicante)

## 2.3. Diseño e implementación del sistema

### 2.3.1. Diagrama de clases de diseño (CU SIGNIFICATIVOS)

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. En un diagrama de clases se pueden distinguir principalmente dos elementos: clases y sus relaciones. (Bula)

### 2.3.2. Diagrama de secuencia (CU Significativos)

### 2.3.3. Diseño de la base de datos

### 2.3.4. Diagrama de componentes de despliegue

## [CONCLUSIONES]

Alicante, U. d. Modelo vista controlador (MVC).

Analysis, I. I. o. B. (2015). *A Guide to the Business Analysis Body of Knowledge(BABOK Guide) v3*.

Baena, L. R. (2013). Fundamentos de la Interacción Persona-Ordenador. *Universidad Pontificia de Salamanca*.

Bula, C. DIAGRAMA DE CLASES EN UML.

Coppola, M. E. (2022, 7 de junio del 2022). ¿Cuál es la estructura HTML de una página web? Retrieved from <https://blog.hubspot.es/website/estructura-html#:~:text=La%20estructura%20HTML%20de%20una%20p%C3%A1gina%20web%20se%20compone%20de,elementos%20visibles%20de%20la%20p%C3%A1gina>.

Cristancho, F. (2022, 24 de marzo del 2022). Retrieved from <https://talently.tech/blog/python-ventajas-y-desventajas/>

Fernández, C. A. (2000). *El Proceso Unificado Rational para el Desarrollo de Software.* Retrieved from <https://www.utm.mx/~caff/doc/El%20Proceso%20Unificado%20Rational.pdf>

Guerrero, J. M. M., & Delgado, C. A. S. (2010). Anexo 2. Ingeniería de Requerimientos.

Mediavilla, E. UML: Modelado de casos de uso.

Murcia, U. d. (2011). Manual Básico de Creación de Páginas Web. *Area de la Tecnologia de la Informacion y las Comunicaciones Aplicadas*.

Néstor Cohen, G. G. R. (2019). *Metodología de la investigación, ¿para qué?: La producción de los datos y los diseños*.

NGuerrero. (2018, 25 mayo). ¿Que Es El Proceso Unificado de Rational (RUP)? Retrieved from <https://www.programaenlinea.net/proceso-unificado-rational-rup/>

Portal Web Universidad Central "Marta Abreu" de Las Villas. (2022, agosto 26). Retrieved from <https://www.uclv.edu.cu/institucion/>

*Proyecto Estratégico 2022*. (2022). Retrieved from La Habana:

Proyecto\_Grado\_UNAD. (2014, 24 de marzo). Metodogía desarrollo software - RUP. Retrieved from <http://metodogiarupgrupo23.blogspot.com/2014/03/fases-metodologia-rup.html>

Rumbaugh, J., Jaconson, I., & Booch, G. (1999). *El lenguaje unificado de modelado manual de*

*referencia*.

Rumbaugh