

UNIVERSIDAD DE MATANZAS
FACULTAD DE CIENCIAS TÉCNICAS
DEPARTAMENTO DE INFORMÁTICA



**Sitio Web de la Intranet corporativa del Parque Científico-
Tecnológico de Matanzas.**

Informe Final de la Investigación

Autor: Stefan Fundora Viera

Tutores: Msc. Sissi Pérez del Pino

Matanzas, Cuba
Diciembre, 2022

Pensamiento.

“Da siempre lo mejor que tienes. Lo que plantes ahora, lo cosecharás más tarde.”

Og Mandino

Declaración de autoría

Yo Stefan Fundora Viera, declaro que este documento es en su totalidad de mi autoría y autorizo a la Universidad de Matanzas Sede Camilo Cienfuegos a su uso en la forma que estime conveniente. Para que así conste firmo la presente a los ____ días del mes de _____ del año 2022.

_____ Firma del Autor

_____ Firma del Tutor

Opinión del tutor del Trabajo de Diploma.

El trabajo de diploma del estudiante Stefan Fundora Viera para optar por el título de ingeniero informático surge a partir de la necesidad del Parque Científico Tecnológico de centralizar toda la información referente a las publicaciones (noticias o documentos) y además el registro de visitantes al lugar, el estudiante mantuvo una comunicación directa con el cliente durante el tiempo del desarrollo de la aplicación por lo que estaba al tanto de los requisitos que debía tener en cuenta y los posibles cambios. Este trabajo fue aumentando en complejidad y la importancia para la entidad. Se pudo dar soluciones a problemas comunes que impedían la mejoría de dichos procesos.

El trabajo de diploma se fue confeccionando a medida que avanzaba el software, se realiza un análisis crítico de los antecedentes y del flujo actual de los procesos, se analizan las tecnologías y herramientas y se determina su justificación de la aplicación. Luego de varias revisiones, durante todo este tiempo, el estudiante demostró dominio de los contenidos, facilidad para adaptarse a los cambios y responsabilidad, lo que se verifica con las calidades del mismo.

Resumen

El Parque Científico Tecnológico de Matanzas, desde ahora PCT, es una infraestructura de apoyo ubicada en la Universidad de Matanzas, con edificaciones y servicios de gestión, para el establecimiento y el desarrollo de empresas basadas en el conocimiento, que operen en un entorno vinculado a centros de investigación y tecnología, así como a instituciones de enseñanza superior. La siguiente investigación surge a partir de la necesidad de proveer al PCT de una herramienta fundamental para la gestión de la información que tributa a la gestión del conocimiento, y para la comunicación interna, que permite mejorar y agilizar la dinámica organizacional. Para la implementación de dicha herramienta se emplearán las siguientes tecnologías de desarrollo de software: PHP, Laravel, y Next como frameworks de Java Script, MySQL como servidor de base de datos y como gestor de base de datos Navicat.

Summary

The Matanzas Science and Technology Park, from now on PCT, is a support infrastructure located at the University of Matanzas, with buildings and management services, for the establishment and development of knowledge-based companies that operate in an environment linked to research and technology centers, as well as higher education institutions. The following research arises from the need to provide the PCT with a fundamental tool for information management that contributes to knowledge management, and for internal communication, which allows improving and streamlining organizational dynamics. For the implementation of this tool, the following software development technologies will be used: PHP, Laravel, and Next as Java Script frameworks, MySQL as a database server and Navicat as a database manager.

Índice

Table of Contents

Introducción	9
1. Capítulo I: Marco Teórico y Tendencias Tecnológicas	13
1.2 Intranet.....	13
1.2.1 Ventajas de la Intranet	13
1.2.1 La intranet y la gestión de información.....	14
1.2.3 La intranet y la comunicación interna	15
1.3 Objeto de estudio	16
1.3.1 Objetivos estratégicos de la organización, misión y visión	16
1.4 Conceptos Asociados al sitio web.....	18
1.5 Metodologías de desarrollo de software.	18
1.5.1 Extreme Programming (XP):	19
1.6 Tecnologías, frameworks y lenguajes de programación	20
1.6.1 Frameworks.....	23
1.6.2 BackEnd	23
1.6.3 FrontEnd.....	24
1.6.4 Servidor de base de datos	25
1.6.5 Herramientas.....	26
1.7 Conclusiones del capítulo.....	27
2. Capítulo II: Análisis, diseño y construcción de la solución propuesta.....	28
2.1 Introducción	28
2.2 Requisitos funcionales	28
2.3 Requisitos no funcionales	29
2.4 Etapa de planificación.....	30
2.5 Equipo y roles de trabajo.....	30
2.6 Historias de Usuarios iniciales.....	31
2.7 Planificación de iteraciones	36
2.8 Etapa de diseño.....	37

2.8.1 Tareas para cada Historia de usuario.....	37
2.9 Conclusiones del capítulo	47
3. Capítulo III: Construcción de la solución propuesta al problema científico.....	48
3.1 Introducción	48
3.2 Pruebas.....	48
3.2.1 Pruebas de aceptación.....	48
3.3 Estudio de factibilidad	55
3.4 Análisis de los resultados obtenidos.....	56
3.5 Conclusiones de capítulo	56
Conclusiones	57
Recomendaciones	58
Bibliografías.....	59
Anexos	62

Introducción

Las intranets son entornos únicos, integrados por personas, cultura corporativa, procesos y objetivos, así como por tecnologías específicas, con un tamaño y recursos determinados; de ahí su amplia variedad. (Cascante & Fonseca, 2006)

Éstas, constituyen una forma de colocar a disposición de los trabajadores todo el potencial de la organización, de forma que les permita resolver problemas, construir nuevos recursos o perfeccionar el empleo de los existentes y divulgar información de manera rápida, así como convertirles en miembros activos de una red corporativa y potenciar entre ellos la comunicación, el intercambio y la colaboración. (Cascante & Fonseca, 2006)

Orientar el desarrollo de la intranet a los procesos principales de la organización y a las actividades de los grupos de trabajo multifuncionales permite mejorar la interacción entre dependencias, funciones y personas separadas o no geográficamente. De esta forma, se constituye como una gran fuente de conocimiento accesible a todos y no meramente un tramitador. (Cascante & Fonseca, 2006)

De forma general puede decirse que la intranet es una alternativa para mejorar y agilizar la dinámica organizacional, al permitir la automatización y mejora de los procesos administrativos y productivos.

Al ser una ventana a la comunicación interna de las entidades y un medio para la difusión de la información, que busca que cada cual cuente con la información necesaria en el momento oportuno sin que tenga que recurrir a terceros para conseguirla, agiliza la toma de decisiones y permite a la organización mantenerse activa en un entorno cambiante y globalizado como el actual. (Cascante & Fonseca, 2006)

El PCT de Matanzas fue creado el 19 de noviembre de 2020 su misión es desarrollar y desplegar en el territorio de la provincia de Matanzas, soluciones TICs (Tecnologías de la Información y las Comunicaciones) integrales, para elevar la calidad de vida y la participación de la población en la gestión de gobierno, la producción y los servicios. [2]

A partir de la integralidad en la gestión del conocimiento, actuar como un puente entre la investigación y el mercado que impulse el desarrollo, la transferencia y la comercialización de la tecnología y la actividad económica sostenible basada en la

innovación, conjugando los recursos económicos e intelectuales mediante la cooperación y las relaciones formales y operativas entre las empresas, el gobierno y el mundo académico, para darle cumplimiento mediante soluciones tecnológicas integrales a la Proyección Estratégica de Informatización Territorial. (UMCC, 2022)

El proceso de gestión del conocimiento y la estrategia comunicacional de la institución se dificulta por las siguientes razones:

1. Fuerte descentralización de la información.
2. Excesiva manipulación de la información lo que provoca posibilidad de errores y una alta carga administrativa.
3. Los datos existentes en ficheros duplicados pueden conducir a error y confusión respecto a sus versiones.
4. Duplicidad de datos, no existe el dato único, mismos ficheros repetidos en las carpetas personales de cada usuario.
5. La fiabilidad del dato es solo garantizada por la profesionalidad del personal.
6. No existe la posibilidad de conocer online el estado de los proyectos, dificultando el conocimiento y las posibles acciones correctoras.
7. Existe poca divulgación de los resultados de las investigaciones de los trabajadores del PCT
8. No se existe la posibilidad de conocer online las opiniones de los visitantes al PCT ni la fecha en que ocurre este proceso.

Todo lo anterior también conlleva a que no exista un sistema efectivo que permita la comunicación entre departamentos, se ralentizan o duplica procesos y, en definitiva, ahoga las posibilidades del talento interno del PCT.

Debido a lo descrito anteriormente se propone como **problema científico** a resolver:

¿Cómo desarrollar una aplicación web para la intranet corporativa del Parque Científico Tecnológico de Matanzas que contribuya al proceso de gestión del conocimiento y la estrategia comunicacional?

Como **Objetivo de Estudio** se identifica la gestión del conocimiento y la estrategia comunicacional.

Campo de acción: La gestión del conocimiento y la estrategia comunicacional del Parque Científico Tecnológico de Matanzas.

El **Objetivo General:** Desarrollar una aplicación web que contribuya al proceso de gestión de conocimiento y la estrategia comunicacional del PCT.

Como **Hipótesis** se plantea que, mediante el desarrollo de una aplicación web para la intranet corporativa del PCT, es posible contribuir a la gestión del conocimiento y a la estrategia comunicacional.

Como **Objetivos Específicos** se tienen los siguientes:

1. Realizar búsqueda bibliográfica para la definición de los conceptos bases referentes al tema de investigación.
2. Analizar los antecedentes referentes al desarrollo de la intranet en Cuba y el mundo.
3. Analizar el flujo actual de los procesos relacionados con el campo de acción.
4. Seleccionar las herramientas y metodologías para el diseño y desarrollo de la aplicación web.
5. Aplicar las etapas de la metodología de desarrollo de software para el desarrollo de la aplicación web para el proceso de gestión del conocimiento y la estrategia comunicacional del PCT.
6. Diseñar e implementar la aplicación web para el proceso de gestión del conocimiento y la estrategia comunicacional del PCT.
7. Validar la solución mediante la realización de pruebas y el análisis del cumplimiento de las especificaciones requeridas por el cliente.

El **resultado de esta investigación** se define en el desarrollo de una aplicación web que permita a los trabajadores del PCT un ambiente fértil para la creación, intercambio y gestión del conocimiento entre los miembros de distintos departamentos, busca que cada cual cuente con la información necesaria en el momento oportuno sin que tenga que recurrir a terceros para conseguirla.

Métodos de investigación utilizados

Los métodos de investigación son las herramientas que los investigadores utilizan para obtener y analizar los datos. Estas incluyen el muestreo, los cuestionarios, las entrevistas, los estudios de casos, el método experimental, los ensayos y grupos de enfoque. (Halcomb & Hickman, 2015)

La elección del método de investigación es predeterminada por el problema a resolver y por los datos que se pueden obtener. Así, tenemos métodos de investigación cuantitativos, cualitativos o mixtos. (Halcomb & Hickman, 2015)

Métodos teóricos:

Método de **análisis histórico – lógico**: permitió estudiar la trayectoria y desarrollo de los sistemas de gestión existentes.

Método de **análisis y síntesis**: este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer lo complejo en sus partes y cualidades, la división del todo en sus múltiples relaciones para luego unir las partes analizadas, descubrir las relaciones y características generales entre ellas.

Método **inductivo - deductivo**: su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se infirieron a partir de propiedades y relaciones existentes entre los elementos que conforman el fenómeno objeto de estudio.

Métodos empíricos:

En esta investigación se utilizaron los métodos de **entrevista** y **observación** con el objetivo de conocer todos los detalles sobre el PCT, el manejo y control de la información y registro de visitantes. Las entrevistas fueron desarrolladas a trabajadores el cual permitió el levantamiento de requisitos y en la comprensión del objeto de estudio, las observaciones a sitios web intranets con el objetivo de nutrirse del conocimiento de los mismos y se realizó el análisis de documentos digitales, relacionados con el tema de investigación. (Telot, 2010)

Esta investigación se estructura de la siguiente forma:

- **Capítulo 1: Marco Teórico Referencial:** se plantean las definiciones fundamentales asociadas al tema de investigación. Se hace un estudio sobre los antecedentes, las tendencias y tecnologías que serán usadas. Se exponen las características fundamentales de los lenguajes de programación, los gestores de bases de datos y las características fundamentales de las metodologías ágiles de desarrollo de software.
- **Capítulo 2: Diseño y Desarrollo de la Solución Propuesta:** se argumenta la solución que se propone al problema de investigación, se presenta una planificación inicial del proyecto, con el empleo de la metodología ágil de desarrollo de software Extreme Programming (XP). Se desarrolla la solución propuesta.
- **Capítulo 3: Validación de la Solución Propuesta:** se realizan pruebas funcionales y se hace un análisis de los resultados obtenidos, basándose en el criterio del cliente y de los demás trabajadores del PCT del desarrollo de software.

1. Capítulo I: Marco Teórico y Tendencias Tecnológicas

1.1 Introducción

En este capítulo se abordan los fundamentos teóricos asociados a la solución del problema descrito en la Introducción, se evidencia el Marco Teórico en donde se muestran los antecedentes de este trabajo, la caracterización del objeto de estudio, se muestran las principales características de la metodología empleada, así como un análisis de las herramientas.

1.2 Intranet

La intranet es una red privada que pertenece a una institución, organización o empresa. Se considera como una red de Internet privada. La intranet en una organización tiene como objetivo conectar, unir, compartir información y datos entre los empleados, estudiantes o personas que pertenecen a dicha organización sin tener que recurrir a un medio público e inseguro. Conecta varios servidores web usando las herramientas de la red de Internet pública como HTTP que es el protocolo de acceso a las páginas web y TCP/IP cuyas siglas son *Transmission control protocol/ Internet protocol* que se traduce al español como el protocolo de control de transmisión/ protocolo Internet. (Portal, 2022)

Intranet Corporativa

Una intranet corporativa es una red privada que permite compartir recursos e información entre los usuarios que formen parte del negocio o colabore con el mismo.

Gracias a sus amplias posibilidades, cumple una importante labor en las empresas en la actualidad. Y no hablamos solo de las grandes corporaciones.

De esta manera, la importancia de la intranet de una empresa reside en que facilita el acceso a la información por parte de los empleados y mejora el trabajo en equipo y la colaboración entre sus integrantes. (Calvo,2022)

1.2.1 Ventajas de la Intranet

1. Para la comunicación:

- Contacto directo entre los miembros de una organización.
- Comunicación entre “expertos”.

2. Para la gestión de la información:

- Edición y publicación rápida de la información interna.
- Eliminación de información innecesaria y redundante.
- Creación de datos críticos.

3. Para la gestión del conocimiento:

- Permite la captura fácil de conocimiento no estructurado e implícito.
- Incrementa el conocimiento y la capacidad de utilizar y compartir los recursos disponibles.
- Promueve una cultura organizativa común.
- Posibilita dibujar el conocimiento colectivo y la experiencia del personal, actual y pretérita.
- Permite tomar decisiones rápidas y bien documentadas con una reducción de los costos.

4. Como infraestructura:

- Reducción de los costos en tecnologías de la información y la comunicación. (Gonzales, 2003)

1.2.1 La intranet y la gestión de información

Con el empleo de la intranet, puede gestionarse la información de la organización relacionada con los recursos humanos, las finanzas, los procedimientos a implantar en la organización, la información financiera interna y hasta noticias de interés para todos los trabajadores.

Brinda, además, la posibilidad de crear y editar publicaciones impresas y en línea, archivos de artículos de publicaciones impresas, reunir información de distintos departamentos para colocarla al alcance de los usuarios, bien para actividades promocionales, de publicidad, de marketing, de ventas y servicio al cliente, o para crear equipos de trabajo, de seguimiento de proyectos, de discusión, entre otros. Permite la organización de la información que posee la institución y la pone a disposición de todo el personal para utilizarla en el momento oportuno para la toma de decisiones. (Pujol, 2007)

La gestión de la información en las organizaciones constituye un factor crítico de éxito, que se acepta unánimemente como recurso indispensable para ampliar la competitividad, aumentar la calidad y la satisfacción de los usuarios, así como para desenvolverse en el ámbito global. Esto ha conducido al desarrollo de sistemas de información para el manejo, tratamiento y uso de la información en las organizaciones que, soportados sobre la intranet e impulsados por los recursos humanos en su condición de agentes y líderes del cambio, posibilitan la agilidad y facilidad de acceso necesaria para ejecución de los procesos organizacionales y la toma de decisiones. (Pujol, 2007)

1.2.3 La intranet y la comunicación interna

En la mayor parte de las organizaciones existen las llamadas *islas de la comunicación*, donde la información valiosa para el desempeño interno reposa indefinidamente en escritorios o computadoras personales, sin compartirse. Todo esto sucede porque no se han apoyado en las TICs para buscar una gestión adecuada de este proceso. (Cortes, 2007)

En este sentido, es importante establecer la diferencia existente entre informar y comunicar. Informar aumenta el conocimiento, se advierten novedades; comunicar implica percepción, creación de expectativas y se plantean exigencias. Informar es parte de la comunicación; lo que se comunica en los mensajes es información, y con esto, la comunicación avanza un paso más en las relaciones entre los empleados, porque ella es capaz de producir determinados comportamientos. Es decir, en la primera se transmite un mensaje; en la segunda se requiere una respuesta del receptor. (Cortes, 2007)

Ejemplos de Intranets

Domino's: Utiliza un servidor para mantener a los usuarios informados de todos los acontecimientos, ofertas, precios o novedades de su franquicia a nivel mundial.

Banca móvil: Las ventajas que ofrece el intranet es el acceso que se puede obtener a través de los teléfonos celulares para gestionar movimientos bancarios, hacer pagos y revisar saldos.

Cines: Además de funcionalidades de trabajo, el intranet permite a una empresa tener contacto directo con los usuarios y así potencializar su desempeño, tal es el caso de los cines en los que se venden entradas por medio de su servidor.

Las aplicaciones de servicio a domicilio de comidas: funcionan por medio de intranet, ya que son servidores exclusivos que requieren de ciertos protocolos para su acceso, pedido y cancelación.

En los procesos electorales: se utiliza el intranet por medio de un programa único y exclusivo en el que solo tiene acceso los expertos en informática, permiten el conteo de votos electrónico. (Liferay, 2022)

1.3 Objeto de estudio y descripción del negocio

1.3.1 Objetivos estratégicos de la organización, misión y visión

Misión: Promover la innovación en las TIC, estimulando y gestionando el flujo del conocimiento y la tecnología, e impulsando la creación, el crecimiento y la competitividad de las empresas.

Visión: somos referentes de la innovación empresarial y el desarrollo de las TIC en el país y en la región de América Latina y el Caribe, transformando proyectos exitosos y conocimientos científicos, en resultados competitivos y sostenibles.

Objetivos estratégicos de la organización:

- Promover el desarrollo científico, tecnológico, la innovación y la transferencia del conocimiento, contribuyendo al desarrollo económico del país y mejorando la competitividad de las empresas.
- Establecer y fortalecer sinergias entre la Universidad, los centros de educación superior, las instituciones de investigación, las empresas y los mercados.
- Impulsar el surgimiento y el crecimiento de empresas basadas en el conocimiento y de otras organizaciones de alto valor añadido.

1.3.2 Descripción y modelado del negocio

El PCT dispone de un proceso de gestión de publicaciones (noticias, documentos, etc.) que no encuentra apoyado por una aplicación informática, este flujo de información se realiza de forma desorganizada y poco segura.

El proceso de gestión de publicaciones comienza cuando un trabajador necesita publicar alguna publicación (noticia o documento), si es de tipo noticia y ésta es pública se envía a un grupo de WhatsApp creado para este fin, si la noticia es privada se le informa personalmente al departamento correspondiente. Si la publicación es de tipo documento, entonces se le entrega personalmente al presidente, quien lo recibe y lo guarda. (Ver Diagrama 1).

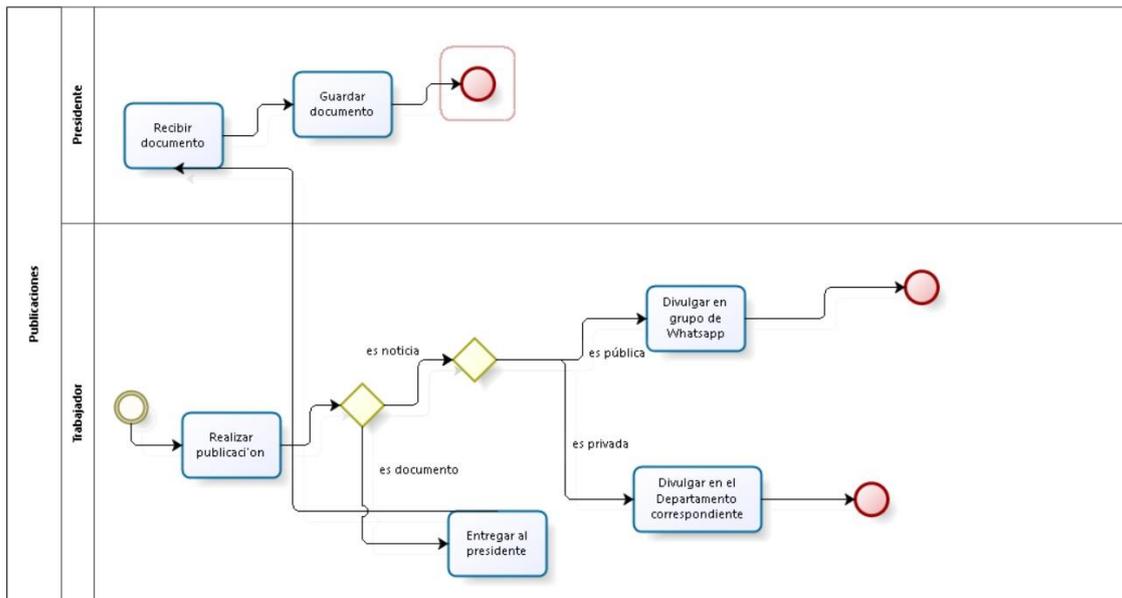


Diagrama 1. Modelado del proceso de Gestionar publicaciones

El proceso de registro de visitantes consiste en que llega el visitante al PCT y realiza una presentación en una fecha determinada y un trabajador la guarda. Cuando se necesita saber en qué momento se recibió una visita específica se recurre a la computadora para buscar el documento, en caso contrario se culmina el proceso. (Ver Diagrama 2)

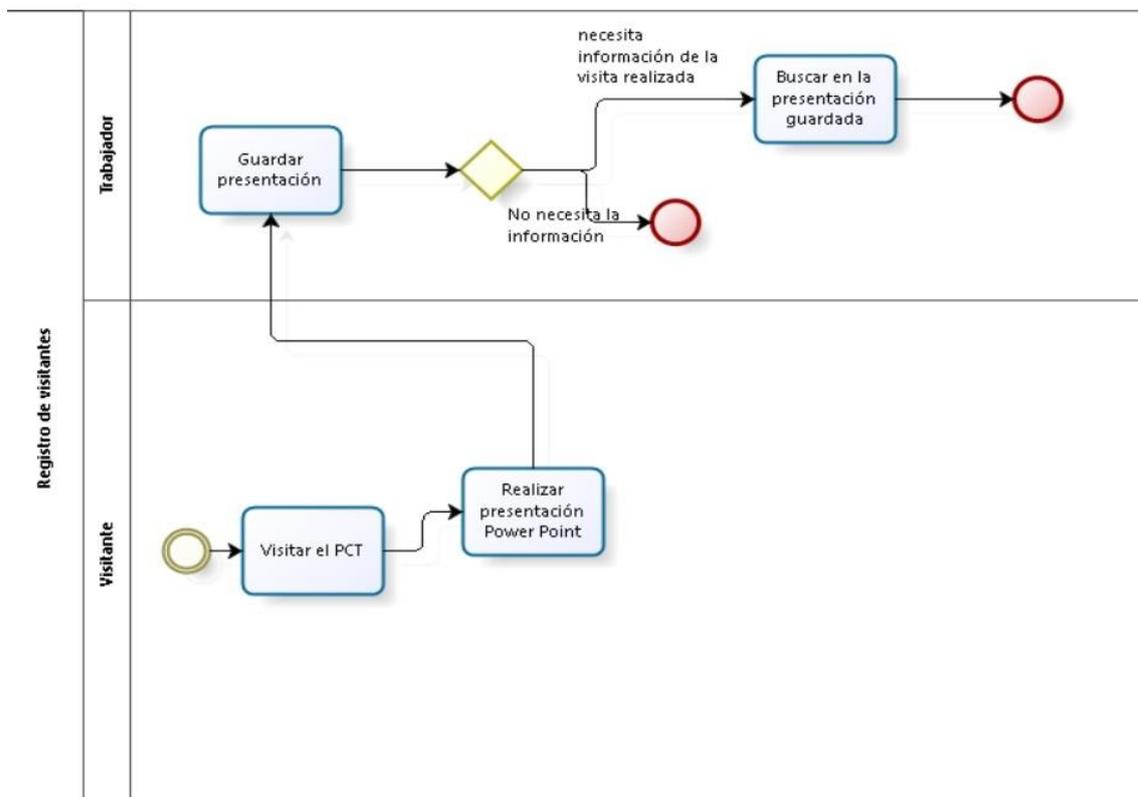


Diagrama 2. Modelado del proceso de registro de visitantes

1.4 Conceptos Asociados al sitio web

Administrador: es aquel trabajador que va a tener todos los permisos para supervisar todo el programa web, gestionando todo. Es decir, este puede crear, eliminar o editar a usuarios, también puede crear o eliminar publicaciones. Este puede ver el sistema de traza de todo el programa web tanto de los usuarios como las publicaciones.

Trabajador: es aquella persona que está registrada en el sistema. Este va a pertenecer a un departamento el cual va a poder crear, editar o eliminar su publicación. También puede ver las publicaciones públicas de otros departamentos y por supuesto a las que perteneces.

Usuario: es aquella persona que no está registrada en el sistema. Este solo podrá ver las publicaciones públicas en se muestran en el Home del programa web.

Publicación: es una información que se desea dar a conocer, ya sea una noticia, un archivo adjunto, un enlace externo o la misma estar compuesta de todo lo mencionado anteriormente. Estas publicaciones pueden o no tener un título, una descripción, una imagen, un link adjunto. Esta la publicación publica que se va a mostrar en el departamento y en el home y la publicación privada que solo se va a ver en el departamento es decir solo la van a ver los trabajadores del departamento al que pertenezca esa publicación.

Noticia: es una información, conocimiento o idea de una cosa.

Archivo Adjunto: es un documento que se envía junto con un mensaje o no, este puede ser un pdf o una imagen, la cual se pueden descargar.

Enlaces Externos: son aquellos que apuntan a un dominio diferente a tu página web, mostrando un contenido específico

Visitantes: son aquellas personas con un alto grado de mando, como el presidente, el viceministro, etc.

Registro de visitantes: es donde se va a poder registrar las visitas, las cuales van a a dejar una opinión acerca del PCT.

1.5 Metodologías de desarrollo de software.

Lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo

umentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. No existe una metodología de software universal, ya que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigiéndose así que el proceso sea configurable. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas. (Pressman, 2015)

Metodologías Ágiles	Metodologías Tradicionales
Pocos Roles, mas genéricos y flexibles	Más Roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones
Menos recursos.	Más recursos.
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos
La arquitectura se va definiendo y mejorando a lo largo del proyecto	Se promueve que la arquitectura se defina tempranamente en el proyecto
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo	Énfasis en la definición del proceso: roles, actividades y artefactos
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Se esperan cambios durante el proyecto	Se espera que no ocurran cambios de gran impacto durante el proyecto

Tabla 1: Comparación Metodología Ágil vs Tradicional. (Fandom, 2014)

Basada en toda esta información adquirida se decide usar las metodologías Ágiles para desarrollar la solución propuesta.

1.5.1 Extreme Programming (XP):

Nace como nueva disciplina de desarrollo de software en manos de Kent Beck. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. (Letelier, 2003)

Los objetivos de XP son muy simples: la satisfacción del cliente, ya que esta metodología trata de dar al cliente el software que necesita y cuando lo necesita, por lo que debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación. El segundo objetivo es potenciar al

máximo el trabajo en grupo, tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software. (Calero, 2003)

Una de las cosas que a los programadores les tiene que quedar muy claro es que, en el ciclo de vida del desarrollo de un proyecto de software, los cambios van a aparecer, pueden cambiar los requisitos, las reglas de negocio, el personal, la tecnología, etc. Por tanto, el problema no es el cambio en sí, ya que este va a suceder sino la incapacidad de enfrentarnos a estos cambios. (Calero, 2003)

Esta metodología tiene como principales características que es inmediata, concreta con continuas vueltas atrás y revisiones en ciclos cortos, incremental, que rápidamente se expande con un plan global y se espera que evolucione durante la vida del proyecto, flexible a la hora de planificar la implementación de funcionalidades, de manera que se adapta con éxito a las necesidades cambiantes. La comunicación oral, las pruebas y el código fuente tienen un fuerte peso, para que se pueda relacionar de manera exacta el sistema con su propósito. (Romero, s.f.)

Luego de investigar sobre las distintas metodologías ágiles dispuestas para el desarrollo del software se define XP como la metodología principal para darle solución el problema de investigación ya que es liviana, ágil y está orientada más a las personas que a los procesos mientras que las metodologías tradicionales están orientados a documentos y se vuelven demasiado burocráticas e ineficaces.

1.6 Tecnologías, frameworks y lenguajes de programación

PHP: es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web. Fue creado inicialmente por el programador danés-canadiense Rasmus Lerdorf en 1994. En la actualidad, la implementación de referencia de PHP es producida por The PHP Group. PHP originalmente significaba *Personal Home Page* (Página personal), pero ahora significa el inicialismo recursivo *PHP: Hypertext Preprocessor*. (Mehdi, Betz, Dovgal, Lopes, Magnusson, Richter, Seguy, Vrana, 2022)

Principales razones para elegir los servicios de desarrollo de PHP para el desarrollo de aplicaciones web:

- Lenguaje libre, el cual puede ser editado fácilmente por cualquier desarrollador o programador.
- Dispone de una sintaxis muy limpia, por lo que la curva de aprendizaje es menor que otros lenguajes de programación.
- Permite crear entornos de trabajo fácilmente.

- Dispone de una instalación de lo más sencilla.
- Se integra de una manera muy simple en la base de datos.
- Existe una gran comunidad, la cual es muy activa, y permite que este en una constante evolución y adaptación a las novedades.
- Es el lenguaje de programación más usado mundialmente.
- PHP es un lenguaje multiplataforma, por lo tanto, se puede usar en diversas aplicaciones y entornos profesionales.
- El código se ejecuta en el lado del servidor, y en el navegador se ejecuta en HTML.
- Actualmente, permite ser usado en un gran número de bases de datos como: Oracle, MySQL, Interbase, SQLite, Sybase, PostgreSQL, SQL Server, entre otras más.
- Ofrece seguridad frente a ataques informáticos llevados a cabo por hackers.
- Dispone de múltiples extensiones, por lo que es uno de los lenguajes usados en proyectos con una alta dificultad.
- Soporta gran cantidad de datos. (Digital Agency, 2022)

JavaScript (JS): es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. (ECMA, 2022)

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (*Server-side JavaScript* o *SSJS*). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. (Domínguez, 2005)

Principales razones para elegir los servicios de desarrollo de JavaScript para el desarrollo de aplicaciones web:

- JavaScript es un lenguaje de programación necesario para el desarrollo de páginas web, pues integra muchas opciones para agregarle valor a tus páginas web. Además, está claro que, en cualquier momento, su uso se hace muy necesario.
- Ahora, una de las ventajas más importantes de JS es que trabaja en la parte del cliente, por lo que ahorra en ancho de banda. Por otro lado, acelera la ejecución del programa de código y del sitio web.

- Ya que una página tendría que funcionar en cualquier tipo de navegador (ya sea Chrome, Firefox o Safari) y eso es permitido gracias a JavaScript, ya que funciona en cualquier tipo de navegador, aunque existen problemas respecto a la lectura. Además, tiene toda la compatibilidad con muchas aplicaciones.
- Dado su popularidad entre el sector de los programadores, hay muchas personas interesadas en compartir su conocimiento sobre el lenguaje de programación, además de presentar sus propios proyectos en repositorios remotos, para el acceso a la información de todo el interesado. (KeepCoding, 2022)

CCS (Cascading Style Sheets): es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. (Clark, 2021)

Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS). (Payne & Corrales, 2022)

HTML (HyperText Markup Language): hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

Es un estándar a cargo del *World Wide Web Consortium* (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW).

Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado. (Gonzales, 2006)

1.6.1 Frameworks

Un framework se podría definir como un entorno de trabajo predispuesto, que posee ciertas herramientas y características útiles que agilizan el desarrollo de un proyecto de programación. Básicamente, el framework facilita la labor del programador. Al trabajar con una base de datos ya conocida puede ahorrar tiempo en desarrollo.

Esto permite que los errores a la hora de desarrollar disminuyan considerablemente, ya que un framework es como una acumulación de bibliotecas. Es decir, no se trabaja con base en comandos de un mismo proyecto, sino en una acumulación de ellos.

Esto, a su vez, promueve un mejor trabajo en equipo en caso de tratarse de empresas de desarrollo. Pero lo más importante, es que el framework sirve para obtener proyectos de desarrollo de mayor calidad al trabajar en una base bien constituida. (Cristancho, 2022)

Principales razones para elegir los servicios de desarrollo de Frameworks para el desarrollo de aplicaciones web:

- No es necesario crear una estructura desde cero.
- Evitar escribir código repetitivo como puede ser accesos a las bases de datos, validaciones de formularios, entre otros.
- Permite acortar los plazos de entrega.
- Garantiza editar el trabajo de una manera más sencilla en el largo plazo, ya que dispone de una base estandarizada.
- Facilita localizar utilidades, librerías, y así adaptarlas al framework. (Digital Agency, 2022)

1.6.2 BackEnd

El backend es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos como, por ejemplo, la comunicación con el servidor. (Arjonilla, 2019)

Laravel

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5, PHP 7 y PHP 8. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC. (Laravel, 2022)

Principales razones para elegir los servicios de desarrollo de Laravel para el desarrollo de aplicaciones web:

- 1) **Rendimiento mejorado:** Laravel proporciona diferentes herramientas para que los desarrolladores mejoren el rendimiento de la aplicación web. El uso de Laravel ayuda a reducir el tiempo para el desarrollo de aplicaciones, no solo esto, sino que también es ventajoso para la adición de nuevas funciones a cualquier aplicación.
- 2) **Alta seguridad:** El uso de Laravel hace que las aplicaciones web sean seguras, ya que no permite que ningún malware o amenaza de seguridad ingrese a las aplicaciones. Eso también significa que su código de desarrollo web está seguro y protegido.
- 3) **Autenticación poderosa:** Laravel facilita a los desarrolladores la creación de aplicaciones web que otorgan acceso a los recursos solo a los usuarios autorizados.
- 4) **Migración de base de datos:** La migración de bases de datos también es una de las principales ventajas de utilizar este marco. Estas migraciones permiten a los desarrolladores deshacer fácilmente los cambios realizados en la base de datos. (Sharma, 2022)

1.6.3 FrontEnd

El frontend es la parte del desarrollo web que se dedica a la parte frontal de un sitio web, en pocas palabras del diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos.

Es esa parte de la página con la que interaccionan los usuarios de la misma, es todo el código que se ejecuta en el navegador de un usuario, al que se le denomina una aplicación cliente, es decir, todo lo que el visitante ve y experimenta de forma directa. (García, 2021)

Next.js

Next.js es un framework de React para desarrollar aplicaciones Javascript de una sola página. Los beneficios de este framework son numerosos, tanto para las aplicaciones de nuestros clientes como para el equipo de desarrollo. Y es que cuanto más interactuamos los usuarios con las webs actuales, más impacientes nos volvemos al no ver cumplidas nuestras expectativas con sitios y aplicaciones que no se cargan en milisegundos. (Studio, 2022)

Principales razones para elegir los servicios de desarrollo de Next.js para el desarrollo de aplicaciones web:

1. **Es ultrarrápido:** la velocidad de Next.js es una de sus principales ventajas. Este rendimiento ultrarrápido reduce el tiempo de creación.
2. **Se puede utilizar rutas API (interfaz de programación de aplicaciones):** ¿Desea utilizar una API de terceros? Next.js facilita esto al ofrecer rutas API. De esa manera, se pueden crear API directamente dentro de la aplicación.
3. **Es altamente personalizable:** A diferencia de Create-React-App, Next.js es fácil de personalizar. Se pueden agregar complementos de Babel y cargadores de Webpack, por ejemplo. (Academy, 2022)

Node.js

Node.js es un entorno de tiempo de ejecución de JavaScript, de ahí su terminación «.js». Este entorno de tiempo es open source, es decir, de código abierto, multiplataforma y que se ejecuta del lado del servidor.

Node.js posee todo lo necesario para ejecutar código JavaScript del lado del servidor. Algo que facilita mucho el trabajo de los desarrolladores y el motivo por el que actualmente sea una de las herramientas de trabajo más usadas en desarrollo web. (Acibeiro, 2022)

1.6.4 Servidor de base de datos

MySQL: es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web. (BD Engines, 2022)

MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. [35]

MySQL es un servidor de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación. (Claburn, 2007)

1.6.5 Herramientas

Visual Studio Code: es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft. (Lardinois, 2015)

Navicat: es una solución ideal para la administración y el desarrollo de MySQL / MariaDB. Permite la conexión a bases de datos MySQL y MariaDB simultáneamente desde una sola aplicación. Este frontal con todo incluido ofrece una intuitiva y potente interfaz gráfica de gran alcance para la gestión, el desarrollo y el mantenimiento de las bases de datos. Ofrece un conjunto completo de herramientas para aquellos que son nuevos en MySQL / MariaDB de manera que sólo necesita su funcionalidad básica o de profesionales como usted. (Aguilar, 2014)

Laragon: es una herramienta para equipos técnicos que permite crear diferentes entornos de desarrollo, facilitando el trabajo con las aplicaciones. Laragon es un programa completísimo y con unas características impresionantes. Es la gran alternativa a XAMPP, pero más completo, aunque si lo queréis adaptar a vuestras preferencias, necesitaréis algunos conocimientos adicionales. Su aprendizaje no es para nada

complicado y merece la pena dedicarle un tiempo, sobre todo, si sois desarrolladores que acabáis de empezar a programar con PHP. (Zúñiga, 2022)

1.7 Conclusiones del capítulo

Se definen conceptos claves para la comprensión del objeto de estudio y del flujo actual de los procesos, estableciendo las bases para la próxima fase de la investigación. Se analizan aplicaciones existentes relacionadas al objeto de investigación para así tener una mayor visión de cómo pueden ser los sitios webs intranets.

La Programación Extrema se selecciona como la metodología adecuada para la planificación y construcción de esta aplicación por poseer características de comunicación directa y amplia con el cliente y buena respuesta ante los cambios.

Se realiza un estudio de las tecnologías y las principales herramientas para el desarrollo de la aplicación web, Laravel para el servidor, Next.js para el cliente, Visual Estudio Code y Navicat.. Se determina MySQL como sistema de gestor de base de datos.

2. Capítulo II: Análisis, diseño y construcción de la solución propuesta

2.1 Introducción

En el presente capítulo se describen los principales elementos de la propuesta utilizando la Metodología de Programación Extrema (XP) y se presenta al equipo de trabajo. De igual forma se realiza el análisis del funcionamiento a través de las Historias de Usuarios (HU) y se muestran las iteraciones; así como el plan de entrega. Se realiza un estudio de factibilidad apoyado en el análisis de costo y beneficios, para determinar si es factible el desarrollo de la aplicación.

2.2 Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Cuando hablamos de las entradas, no necesariamente hablamos sólo de las entradas de los usuarios. Pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos. (Blog, 2018)

A continuación, se muestran los siguientes requisitos funcionales del software:

1. Diseño y creación de la interfaz de usuario
2. Diseño y creación de la base de datos
3. Autenticarse
4. Registrarse
5. Gestionar usuarios
 - 5.1 Crear
 - 5.2 Editar
 - 5.3 Listar
 - 5.4 Eliminar
6. Gestionar visitas
 - 6.1 Crear
 - 6.2 Editar
 - 6.3 Listar
 - 6.4 Eliminar
7. Gestionar publicaciones
 - 7.1 Crear
 - 7.2 Editar
 - 7.3 Listar
 - 7.4 Eliminar

8. Gestionar departamentos
 - 8.1 Crear
 - 8.2 Editar
 - 8.3 Listar
 - 8.4 Eliminar
9. Gestionar permisos
 - 9.1 Asignar
 - 9.2 Listar
10. Gestionar trazas de seguridad
 - 10.1 Listar
11. Gestionar notificaciones
 - 11.1 Listar

2.3 Requisitos no funcionales

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace. Alternativamente, definen restricciones del sistema tales como la capacidad de los dispositivos de entrada/salida y la representación de los datos utilizados en la interfaz del sistema. (Blog, 2018)

Los requisitos no funcionales se originan en la necesidad del usuario, debido a restricciones presupuestarias, políticas organizacionales, la necesidad de interoperabilidad con otros sistemas de software o hardware, o factores externos tales como regulaciones de seguridad, políticas de privacidad, entre otros. (Blog, 2018)

- El sistema podrá instalarse sobre Windows, Linux o Unix de forma tal que no haya dificultades en cambiar, de una a otra plataforma, sin necesidad de efectuar cambios.
- El diseño de la interfaz debe ser agradable, ágil, simple de usar. Se debe mostrar un contenido legible y confiable.
- El sistema podrá ser usado por cualquier persona que posea los conocimientos básicos en el manejo de la computadora y el ambiente Web en sentido general.
- El sistema creado es una aplicación Web, entonces el tiempo de respuesta debe ser lo más cercano posible al tiempo real, se necesita un alto grado de eficiencia y un tiempo de respuesta muy breve, para lograr un incremento de la productividad.

- Las contraseñas de los usuarios se deben encriptar, para almacenarla en la base de datos, pues de esta manera se asegura que obtenerlas en texto claro, sea más difícil o quizás imposible.

2.4 Etapa de planificación

Esta etapa tiene como objetivo lograr una eficiente organización de la formulación inicial del problema existente, para brindar una solución. Para realizar la planificación se crean las Historias de Usuarios (HU), a partir de las ideas del cliente obteniendo un punto de partida para el resto del desarrollo del proyecto. Se estima las entregas y el tiempo necesarios teniendo en cuenta retrasos inesperados o cambios que podrían ocurrir durante su confección.

2.5 Equipo y roles de trabajo

De acuerdo a la metodología, la conformación del equipo de trabajo puede variar dependiendo del proyecto que se esté ejecutando. Se reafirma el principio de 40 horas de trabajo semanales, un tiempo prudencial de descanso cada dos o tres aproximadamente. La simplicidad es la base de la programación extrema.

Se simplifica el diseño para agilizar el desarrollo y facilitar el mantenimiento. Unos diseños complejos del código junto a sucesivas modificaciones por parte de diferentes desarrolladores hacen que la complejidad aumente exponencialmente.

La programación debe ser en pequeñas versiones poco a poco, en caso de algún fallo se efectúa la reprogramación del código sin que presente problemas la funcionalidad inicial. Es fundamental la retroalimentación con el cliente, al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real.

Tabla 2: Equipo de trabajo y roles.

Miembros	Roles
Stefan Fundora Viera	Programador, Encargado de Pruebas, Encargado de Seguimiento
Sissi Pérez del Pino	Entrenador, Gestor, Consultor
Directora de Investigación y Desarrollo del Parque Científico Tecnológico	Cliente, encargado de Pruebas

Fuente: Elaboración propia.

2.6 Historias de Usuarios iniciales

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Una historia de usuario es una descripción breve, informal y en lenguaje sencillo de lo que un usuario quiere hacer dentro de un producto de software para obtener algo que le resulte valioso. (Digite, 2021)

En la Tabla 3 se muestra un resumen de todas las HU planificadas inicialmente, en las que queda definido el nivel de prioridad con el que deben darle solución a las HU (P), el riesgo en desarrollo (R). Se define también los puntos de estimación del esfuerzo (E) requerido, que no es más que el tiempo en el que se concibió inicialmente el desarrollo de cada HU.

No	Nombre	Prioridad (P)	Riesgo (R)	Iteraciones	Ptos estimados (E)
1	Diseño de la interfaz de usuario.	Alta	Alta	1	1
2	Diseño de la base de datos	Alta	Alta	1	1
3	Autenticarse	Alta	Alta	2	0.5
4	Registrarse	Alta	Alta	2	0.5
5	Gestionar usuarios.	Alta	Medio	3	1
6	Gestionar visitas	Alta	Medio	3	2
7	Gestionar publicaciones	Alta	Alta	4	3
8	Gestionar departamentos	Alta	Alta	4	3
9	Gestionar permisos	Alta	Alta	5	3
10	Gestionar Trazas de Seguridad	Alta	Alta	6	2
11	Gestionar Notificaciones	Alta	Alta	7	3
Totales				7	20

Tabla 3: Resumen de historias de usuario.

A continuación, se muestran las descripciones de HU:

Tabla 4: Historia de Usuario 1

Historia de usuario	
Número: 1	Usuario: todos
Nombre de Historia: Diseño de la interfaz de usuario	

Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando el programador realiza el diseño de la interfaz de usuario y su creación.	

Fuente: Elaboración propia.

Tabla 5: Historia de Usuario 2

Historia de usuario	
Número: 2	Usuario: todos
Nombre de Historia: Diseño de la base de datos	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando el programador realiza el diseño de la base de datos y su creación. (ver anexo1)	

Fuente: Elaboración propia.

Tabla 6: Historia de Usuario 3

Historia de usuario	
Número: 3	Usuario: todos
Nombre de Historia: Autenticarse	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando el trabajador o el administrador intentan acceder al sistema.	

Fuente: Elaboración propia.

Tabla 7: Historia de Usuario 4

Historia de usuario	
Número: 4	Usuario: todos
Nombre de Historia: Registrarse	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 2
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando un usuario le pide al administrador que le cree las credenciales.	

Fuente: Elaboración propia.

Tabla 8: Historia de Usuario 5

Historia de usuario	
Número: 5	Usuario: Administrador
Nombre de Historia: Gestionar usuarios	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando el administrador decide crear, editar, listar o eliminar a un usuario	

Fuente: Elaboración propia.

Tabla 9: Historia de Usuario 6

Historia de usuario	
Número: 6	Usuario: Administrador
Nombre de Historia: Gestionar visitas	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Stefan Fundora Viera	

<p>Descripción:</p> <p>Se inicia cuando el administrador decide crear, editar, listar o eliminar las visitar.</p>
--

Fuente: Elaboración propia.

Tabla 10: Historia de Usuario 7

Historia de usuario	
Número: 7	Usuario: Administrador y trabajadores
Nombre de Historia: Gestionar publicaciones	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Stefan Fundora Viera	
<p>Descripción:</p> <p>Se inicia cuando el trabajador o administrador decide crear, editar, eliminar sus publicaciones. El administrador puede eliminar las publicaciones de los trabajadores.</p>	

Fuente: Elaboración propia.

Tabla 11: Historia de Usuario 8

Historia de usuario	
Número: 8	Usuario: Administrador
Nombre de Historia: Gestionar departamentos	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Stefan Fundora Viera	
<p>Descripción:</p> <p>Se inicia cuando el administrador decide crear, editar o eliminar departamentos.</p>	

Fuente: Elaboración propia.

Tabla 12: Historia de Usuario 9

Historia de usuario	
Número: 9	Usuario: Administrador

Nombre de Historia: Gestionar permisos	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 5
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando el administrador les asigna permisos específicos a los trabajadores.	

Fuente: Elaboración propia.

Tabla 13: Historia de Usuario 10

Historia de usuario	
Número: 10	Usuario: Administrador
Nombre de Historia: Gestionar trazas de seguridad	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 2	Iteración asignada: 6
Programador responsable: Stefan Fundora Viera	
Descripción: El administrador es el único que puede ver las trazas de los usuarios, departamentos y publicaciones.	

Fuente: Elaboración propia.

Tabla 13: Historia de Usuario 11

Historia de usuario	
Número: 11	Usuario: todos
Nombre de Historia: Gestionar notificaciones	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 7
Programador responsable: Stefan Fundora Viera	
Descripción: Se inicia cuando al administrador le llegan por parte de los usuarios la petición de la creación de las credenciales.	

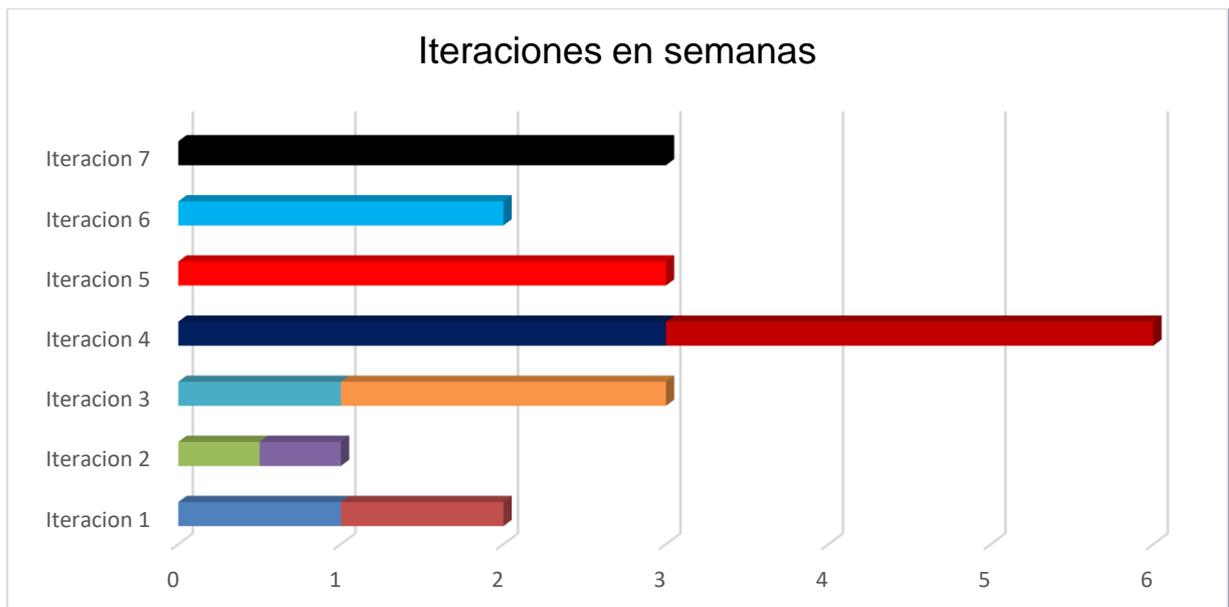
Fuente: Elaboración propia.

2.7 Planificación de iteraciones

Las funcionalidades son desarrolladas en esta fase y se genera al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

En la siguiente tabla se muestran las iteraciones y la cantidad de semanas que se demoraron en realizar cada tarea.

Tabla 14: Planificación de las iteraciones en semanas



Fuente: Elaboración propia.

Leyenda:

- Iteración 1: Diseño de la interfaz de usuario, Diseño de la base de datos.
- Iteración 2: Autenticarse, Registrarse.
- Iteración 3: Gestionar usuarios, Gestionar visitas.
- Iteración 4: Gestionar publicaciones, Gestionar departamentos.
- Iteración 5: Gestionar permisos, Gestionar trazas de seguridad.
- Iteración 5: Gestionar notificaciones.

2.8 Etapa de diseño

En XP solo se diseñan aquellas historias de usuario que el cliente ha seleccionado para la iteración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en las fases iniciales del proyecto para luego modificarlo, se considera un desperdicio de tiempo.

Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo de un diseño inicial que va siendo corregido y mejorado en el transcurso del proyecto.

2.8.1 Tareas para cada Historia de usuario

La tabla 15 muestra la cantidad de tareas correspondientes a cada HU.

Tabla 15. Tareas de Iteración a cada HU

No	Nombre HU	No	Tarea de Iteración	Iteraciones
1	Diseño de la interfaz de usuario	1	Diseñar	1
		2	Crear	
2	Diseño de la base de datos	3	Diseñar	1
		4	Crear	
3	Autenticarse	5	Crear	2
		6	Validar	
4	Registrarse	7	Crear	2
		8	Validar	
5	Gestionar usuarios	9	Crear	3
		10	Editar	
		11	Listar	
		12	Eliminar	
6	Gestionar visitas	13	Crear	3
		14	Editar	
		15	Listar	
		16	Eliminar	
7	Gestionar publicaciones	17	Crear	4
		18	Editar	

		19	Listar	
		20	Eliminar	
8	Gestionar departamentos	21	Crear	4
		22	Edita	
		23	Listar	
		24	Eliminar	
9	Gestionar permisos	25	Asignar	5
		26	Listar	
10	Gestionar trazas de seguridad	27	Listar	6
11	Gestionar notificaciones	28	Listar	7

Fuente: Elaboración propia.

A continuación, se muestran las relaciones de las tareas de iteración a realizar durante el desarrollo de este proyecto:

Tabla 16. Tareas de Iteración 1

Tarea de iteración	
Número: 1	No Historia: 1
Nombre de Tarea: Diseñar interfaz	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz de usuario.	

Fuente: Elaboración propia.

Tabla 17. Tareas de Iteración 2

Tarea de iteración	
Número: 2	No Historia: 1
Nombre de Tarea: Crear interfaz	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.8
Programador responsable: Stefan Fundora Viera	
Descripción: Después de diseñada la interfaz de usuario se empieza a crear en el sistema.	

Fuente: Elaboración propia.

Tabla 18. Tareas de Iteración 3

Tarea de iteración	
Número: 3	No Historia: 2
Nombre de Tarea: Diseñar la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la base de datos	

Fuente: Elaboración propia.

Tabla 19. Tareas de Iteración 4

Tarea de iteración	
Número: 4	No Historia: 2
Nombre de Tarea: Crear la base de datos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Stefan Fundora Viera	
Descripción: Después de diseñada se crean las clases y la base de datos en MySQL Server	

Fuente: Elaboración propia.

Tabla 20. Tareas de Iteración 5

Tarea de iteración	
Número: 5	No Historia: 3
Nombre de Tarea: Crear sistema de autenticación	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para introducir los datos de acceso. Después de verificarlos en la base de datos se procede a autenticar al usuario en el sistema.	

Fuente: Elaboración propia.

Tabla 21. Tareas de Iteración 6

Tarea de iteración	
--------------------	--

Número: 6	No Historia: 3
Nombre de Tarea: Validar información de acceso	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.3
Programador responsable: Stefan Fundora Viera	
Descripción: Validar la información obtenida en los formularios de entrada	

Fuente: Elaboración propia.

Tabla 22. Tareas de Iteración 7

Tarea de iteración	
Número: 7	No Historia: 4
Nombre de Tarea: Crear sistema de registro	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña una interfaz con un formulario para solicitar al administrador que le cree las credenciales para poder autenticarse en el sistema.	

Fuente: Elaboración propia.

Tabla 23. Tareas de Iteración 8

Tarea de iteración	
Número: 8	No Historia: 4
Nombre de Tarea: Validar información de acceso	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.3
Programador responsable: Stefan Fundora Viera	
Descripción: Validar la información obtenida en los formularios de entrada.	

Fuente: Elaboración propia.

Tabla 24. Tareas de Iteración 9

Tarea de iteración	
Número: 9	No Historia: 5
Nombre de Tarea: Crear usuario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.3
Programador responsable: Stefan Fundora Viera	

<p>Descripción:</p> <p>Se diseña la interfaz para introducir los datos del nuevo trabajador. Después que se verifiquen los datos se selecciona la opción de crear y se muestra una tabla actualizada con los que existe en la base de datos. Solo el administrador puede acceder a esta opción.</p>
--

Fuente: Elaboración propia.

Tabla 25. Tareas de Iteración 10

Tarea de iteración	
Número: 10	No Historia: 5
Nombre de Tarea: Editar usuario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.3
Programador responsable: Stefan Fundora Viera	
<p>Descripción:</p> <p>Se diseña la interfaz para editar los datos del trabajador seleccionado. Después de verificar los datos se selecciona la opción de insertar y se muestra una tabla con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.</p>	

Fuente: Elaboración propia.

Tabla 26. Tareas de Iteración 11

Tarea de iteración	
Número: 11	No Historia: 5
Nombre de Tarea: Listar usuario	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
<p>Descripción:</p> <p>Se pueden listar todos los usuarios registrados en la base de datos. Solo el administrador puede acceder a esta opción.</p>	

Fuente: Elaboración propia.

Tabla 27. Tareas de Iteración 12

Tarea de iteración	
Número: 12	No Historia: 5
Nombre de Tarea: Eliminar usuario	

Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para eliminar a un usuario seleccionado. Se muestra una tabla actualizada con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 28. Tareas de Iteración 13

Tarea de iteración	
Número: 13	No Historia: 6
Nombre de Tarea: Crear visitante	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz con un formulario para registrar al visitante. Cuando se registre se va a mostrar una tabla actualizada con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 29. Tareas de Iteración 14

Tarea de iteración	
Número: 14	No Historia: 6
Nombre de Tarea: Editar visitante	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para editar los datos del visitante seleccionado. Después de verificar los datos se selecciona la opción de insertar y se muestra una tabla con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 30. Tareas de Iteración 15

Tarea de iteración	
Número: 15	No Historia: 6
Nombre de Tarea: Listar visitante	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se pueden listar todas las visitas registrados en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 31. Tareas de Iteración 16

Tarea de iteración	
Número: 16	No Historia: 6
Nombre de Tarea: Eliminar visitantes	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.3
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para eliminar a un visitante seleccionado. Se muestra una tabla actualizada con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 32. Tareas de Iteración 17

Tarea de iteración	
Número: 17	No Historia: 7
Nombre de Tarea: Crear publicación	
Tipo de Tarea: Desarrollo	Puntos estimados: 1.5
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña una interfaz con un formulario para crear una publicación. Pueden acceder a esta opción los trabajadores y el administrador.	

Fuente: Elaboración propia.

Tabla 33. Tareas de Iteración 18

Tarea de iteración

Número: 18	No Historia: 7
Nombre de Tarea: Editar publicación	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña una interfaz con un formulario para editar una publicación. Pueden acceder a esta opción los trabajadores y el administrador, pero solo podrán editar sus publicaciones pertenecientes.	

Fuente: Elaboración propia.

Tabla 34. Tareas de Iteración 19

Tarea de iteración	
Número: 19	No Historia: 7
Nombre de Tarea: Listar publicación	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se pueden listar las publicaciones. Pueden acceder a esta opción los trabajadores y el administrador.	

Fuente: Elaboración propia.

Tabla 35. Tareas de Iteración 20

Tarea de iteración	
Número: 20	No Historia: 7
Nombre de Tarea: Eliminar publicación	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.3
Programador responsable: Stefan Fundora Viera	
Descripción: Se pueden eliminar las publicaciones. Pueden acceder a esta opción los trabajadores y el administrador, pero solo el trabajador podrá eliminar sus publicaciones pertenecientes, el administrador puede borrar tanto sus publicaciones como la de los trabajadores.	

Fuente: Elaboración propia.

Tabla 36. Tareas de Iteración 21

Tarea de iteración	
Número: 21	No Historia: 8

Nombre de Tarea: Crear departamentos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para introducir los datos del nuevo departamento. Después que se verifiquen los datos se selecciona la opción de crear y se muestra una tabla actualizada con los que existe en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 37. Tareas de Iteración 22

Tarea de iteración	
Número: 22	No Historia: 8
Nombre de Tarea: Editar departamentos	
Tipo de Tarea: Desarrollo	Puntos estimados: 1
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para editar los datos del departamento seleccionado. Después de verificar los datos se selecciona la opción de insertar y se muestra una tabla con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 38. Tareas de Iteración 23

Tarea de iteración	
Número: 23	No Historia: 8
Nombre de Tarea: Listar departamentos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Stefan Fundora Viera	
Descripción: Se pueden listar los departamentos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 39. Tareas de Iteración 24

Tarea de iteración

Número: 24	No Historia: 8
Nombre de Tarea: Eliminar departamentos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.5
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz para eliminar al departamento seleccionado. Se muestra una tabla actualizada con los que existen en la base de datos. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 40. Tareas de Iteración 25

Tarea de iteración	
Número: 25	No Historia: 9
Nombre de Tarea: Asignar permisos	
Tipo de Tarea: Desarrollo	Puntos estimados: 2.8
Programador responsable: Stefan Fundora Viera	
Descripción: A medida que se vas creando los trabajadores se les va asignando permisos correspondientes. Solo el administrador puede acceder a esta opción.	

Fuente: Elaboración propia.

Tabla 41. Tareas de Iteración 26

Tarea de iteración	
Número: 26	No Historia: 9
Nombre de Tarea: Listar permisos	
Tipo de Tarea: Desarrollo	Puntos estimados: 0.2
Programador responsable: Stefan Fundora Viera	
Descripción: Se puede listar los permisos a asignar a los trabajadores.	

Fuente: Elaboración propia.

Tabla 42. Tareas de Iteración 27

Tarea de iteración	
Número: 27	No Historia: 10

Nombre de Tarea: Listar trazas de seguridad	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Stefan Fundora Viera	
Descripción: Se diseña la interfaz donde se va a mostrar y listar todas las trazas correspondientes a los usuarios, los departamentos y las publicaciones	

Fuente: Elaboración propia.

Tabla 43. Tareas de Iteración 28

Tarea de iteración	
Número: 28	No Historia: 11
Nombre de Tarea: Listar notificaciones	
Tipo de Tarea: Desarrollo	Puntos estimados: 2
Programador responsable: Stefan Fundora Viera	
Descripción: Al administrador le va a llegar una lista de peticiones de creación de credenciales de todo aquel usuario que quiera registrarse en el sistema. Mediante el sistema de notificaciones le va hacer llegar las credenciales a ese usuario a través del correo @gmail.com	

Fuente: Elaboración propia.

2.9 Conclusiones del capítulo

Una vez descrita la propuesta de solución al problema científico de esta investigación se llegó a la conclusión de que se dispone de un equipo de trabajo capaz de solucionar el problema existente.

La planificación inicial es necesaria para tener una visión general del problema que se enfrenta y pensar desde el inicio en soluciones eficaces. Se define el plan de iteraciones y el plan de entregas a seguir, siempre teniendo en cuenta que las mismas pueden sufrir modificaciones durante el proceso de implementación.

Historias de usuarios permitieron especificar los requisitos funcionales para que el software tenga una buena aceptación, siendo la comunicación con el cliente el factor más importante.

Después de esta planificación inicial se puede comenzar a construir el sistema que constituye la propuesta de solución.

3. Capitulo III: Construcción de la solución propuesta al problema científico.

3.1 Introducción

En este capítulo se comprueba el cumplimiento de los requerimientos iniciales, con este propósito se desarrollan los casos de pruebas al software permitiendo la calidad de este producto, lo que constituye uno de los pasos más importantes en el diseño e implementación de un sistema. No debe existir ninguna característica en el programa que no haya sido probada con la intención de buscar errores en el programa y darle solución. Se presentan las pruebas que le dan validez al sistema y un estudio de factibilidad.

3.2 Pruebas

Las pruebas del software son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su propuesta en producción. La interesante de las pruebas es que se puedan ejecutar de manera automática, para determinar en cualquier momento si tenemos una aplicación estable o si, por el contrario, un cambio en una parte ha afectado a otras partes sin que nos demos cuenta. (Turrado, 2020)

3.2.1 Pruebas de aceptación

Las pruebas de aceptación son las realizadas por el cliente y usuarios finales de la aplicación. Permiten probar las funcionalidades que exige el cliente. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue dentro del proyecto.

Las pruebas persiguen como objetivo, llevar a cabo el proceso de ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar al menos un error no descubierto hasta entonces. (Pressman, 2015).

Tabla 44. Prueba de aceptación 1

Pruebas de aceptación	
Número Caso de Prueba: 1	No Historia:1
Nombre Caso de Prueba: Test Interfaz Principal	
Descripción: Verificar que se muestren las interfaces visuales implementadas	

Condiciones de ejecución: Esté corriendo la aplicación.
Entradas: Interfaces de la aplicación.
Resultado esperado: Se muestren las interfaces visuales de la aplicación.
Evaluación: Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 45. Prueba de aceptación 2

Pruebas de aceptación	
Número Caso de Prueba: 2	No Historia: 2
Nombre Caso de Prueba: Crear la base de datos	
Descripción: Se comprueba que haya sido creada correctamente la base de datos	
Condiciones de ejecución: Solo pueden realizar las acciones crear, editar o eliminar aquel trabajador con los permisos adquiridos	
Entradas: El administrador realiza los siguientes pasos: <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar datos de manera incorrecta. 4. Editar los datos dejando campos en blanco. 5. Editar los datos correctamente. 6. Verificar que se muestren los datos actualizados. 	
Resultado esperado: El sistema debe alertar al usuario cuando se inserten datos de manera incorrecta. Cuando se inserten los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos. Cuando se editen los datos correctamente, el sistema debe almacenarlos en la base de datos y mostrarlos actualizados.	
Evaluación: Prueba satisfactoria	

Fuente: Elaboración propia.

Tabla 46. Prueba de aceptación 3

Pruebas de aceptación	
Número Caso de Prueba: 3	No Historia: 3
Nombre Caso de Prueba: Autenticación	
Descripción: Verificar que se autentifique un usuario correctamente en el sistema.	
Condiciones de ejecución: Que el sistema esté conectado a la base de datos	
Entradas: <ol style="list-style-type: none"> 1. Insertar nombre de usuario y la contraseña. 2. Insertar datos incorrectamente 3. Dejar campos en blancos y tratar de acceder 	
Resultado esperado: El sistema debe alertar al usuario cuando se inserten datos erróneos. Cuando se inserten los datos correctamente, el usuario debe entrar y hacer uso de las acciones que le están permitidas según los permisos asociados a él. (ver anexos 2)	
Evaluación: Prueba satisfactoria	

Fuente: Elaboración propia.

Tabla 47. Prueba de aceptación 4

Pruebas de aceptación	
Número Caso de Prueba: 4	No Historia: 4
Nombre Caso de Prueba: Solicitar Registro	
Descripción: El usuario inserta los datos en el formulario para solicitarle al administrador que le cree las credenciales.	
Condiciones de ejecución: Que el sistema esté conectado a la base de datos	
Entradas: <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar datos de manera incorrecta. 	
Resultado esperado: El sistema debe alertar al usuario cuando se inserten datos erróneos. Cuando se inserten los datos correctamente el sistema debe llegar a saber al administrador la solicitud de registro con los datos del usuario.	

Evaluación:

Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 48. Prueba de aceptación 5

Pruebas de aceptación	
Número Caso de Prueba: 5	No Historia: 5
Nombre Caso de Prueba: Usuario	
Descripción: Verificar que se puede crear, editar, listar o eliminar usuarios.	
Condiciones de ejecución: Solo puede realizar estas acciones de crear, editar, listar o eliminar usuarios el administrador o aquel que cuente con los permisos para realizarlos.	
Entradas: <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar datos de manera incorrecta. 4. Editar los datos dejando campos en blanco. 5. Editar los datos de forma correcta. 6. Listar usuarios. 7. Eliminar usuario. 	
Resultado esperado: El sistema debe alertar cuando se inserten datos erróneos. Cuando se inserten los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos. Cuando se editen los datos correctamente el sistema debe almacenarlos en la base de datos y actualizarlos. Se pueden listar los usuarios. Al eliminar debe mostrar una alerta de confirmación para proceder a su eliminación. (ver anexo 3)	
Evaluación: Prueba satisfactoria	

Fuente: Elaboración propia.

Tabla 49. Prueba de aceptación 6

Pruebas de aceptación	
Número Caso de Prueba: 6	No Historia: 6
Nombre Caso de Prueba: Visitas	
Descripción: Verificar que se puede crear, editar, listar o eliminar visitas.	

<p>Condiciones de ejecución:</p> <p>Solo puede realizar estas acciones de crear, editar, listar o eliminar visitas el administrador.</p>
<p>Entradas:</p> <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar datos de manera incorrecta. 4. Editar los datos dejando campos en blanco. 5. Editar los datos de forma correcta. 6. Listar visitas. 7. Eliminar visitas.
<p>Resultado esperado:</p> <p>El sistema debe alertar cuando se dejen campos obligatorios en blanco. Cuando se inserten los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos. Cuando se editen los datos correctamente el sistema debe almacenarlos en la base de datos y actualizarlos. Se pueden listar los visitantes. Al eliminar debe mostrar una alerta de confirmación para proceder a su eliminación.(ver anexos 4)</p>
<p>Evaluación:</p> <p>Prueba satisfactoria</p>

Fuente: Elaboración propia.

Tabla 50. Prueba de aceptación 7

Pruebas de aceptación	
Número Caso de Prueba: 7	No Historia: 7
Nombre Caso de Prueba: Publicaciones	
Descripción: Verificar que se puede crear, editar, listar o eliminar publicaciones.	
Condiciones de ejecución: Las acciones de crear, editar, listar o eliminar publicaciones las podrán hacer siempre tanto el trabajador como el administrador siempre que sea sus publicaciones. El administrador también puede eliminar publicaciones de otro trabajador.	
Entradas: <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar datos de manera incorrecta. 4. Cambiar de tipo de publicación pública o privada 	

<ol style="list-style-type: none"> 5. Editar los datos de la publicación dejando campos en blanco. 6. Editar los datos de la publicación de forma correcta. 7. Listar publicaciones. 8. Eliminar publicaciones.
<p>Resultado esperado:</p> <p>El sistema debe alertar cuando se dejen campos obligatorios en blanco o haya datos erróneos. Cuando se inserten los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos. Cuando se editen los datos correctamente el sistema debe almacenarlos en la base de datos y actualizarlos. Se pueden listar las publicaciones. Se puede elegir como quieren que sea la publicación, pública o privada. Al eliminar debe mostrar una alerta de confirmación para proceder a su eliminación.(ver anexo 5)</p>
<p>Evaluación:</p> <p>Prueba satisfactoria</p>

Fuente: Elaboración propia.

Tabla 51. Prueba de aceptación 8

Pruebas de aceptación	
Número Caso de Prueba: 8	No Historia: 8
Nombre Caso de Prueba: Departamentos	
Descripción: Verificar que se puede crear, editar, listar o eliminar departamentos.	
Condiciones de ejecución: Solo puede realizar estas acciones de crear, editar, listar o eliminar departamentos el administrador.	
Entradas: <ol style="list-style-type: none"> 1. Dejar campos en blanco. 2. Insertar los datos correctamente. 3. Insertar datos de manera incorrecta. 4. Editar los datos dejando campos en blanco. 5. Editar los datos de forma correcta. 6. Listar departamentos. 7. Eliminar departamentos. 	
Resultado esperado: El sistema debe alertar cuando se dejen campos obligatorios en blanco o haya datos erróneos. Cuando se inserten los datos correctamente el sistema debe almacenarlos en la base de datos y mostrarlos. Cuando se editen los datos correctamente el sistema debe almacenarlos en la base de datos y actualizarlos. Se pueden listar los departamentos. Al	

eliminar un departamento debe mostrar una alerta de confirmación para proceder a su eliminación.

Evaluación:

Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 52. Prueba de aceptación 9

Pruebas de aceptación	
Número Caso de Prueba: 9	No Historia:9
Nombre Caso de Prueba: Asignar permisos	
Descripción: Verificar que se puedan asignar los permisos correctamente a los usuarios cuando se estén creando.	
Condiciones de ejecución: Solo puede realizar esta acción de asignar permiso el administrador.	
Entradas: <ol style="list-style-type: none">1. Listar permisos.2. Asignar permisos correspondientes.	
Resultado esperado: El sistema debe asignar permisos correspondientes con el trabajador. El trabajador es capaz de trabajar en el sistema y realizar toda aquella acción relacionada con los permisos asignados (ver anexos 3)	
Evaluación: Prueba satisfactoria	

Fuente: Elaboración propia.

Tabla 53. Prueba de aceptación 10

Pruebas de aceptación	
Número Caso de Prueba: 10	No Historia:10
Nombre Caso de Prueba: Trazas de seguridad	
Descripción: Verificar que en el sistema se guarden las trazas correspondientes a las acciones que se hayan realizado ya se tanto crear, editar, eliminar usuarios, departamentos, publicaciones.	
Condiciones de ejecución: El sistema de trazas de seguridad solo la podrá ver el administrador.	

Entradas: <ol style="list-style-type: none"> 1. Crear, editar, eliminar usuarios. 2. Crear, editar, eliminar publicaciones. 3. Crear, editar, eliminar departamentos.
Resultado esperado: El sistema es capaz de guardar toda aquella acción descrita anteriormente en el sistema de trazas de seguridad. (ver anexos 6)
Evaluación: Prueba satisfactoria

Fuente: Elaboración propia.

Tabla 54. Prueba de aceptación 11

Pruebas de aceptación	
Número Caso de Prueba: 11	No Historia: 11
Nombre Caso de Prueba: Sistema de notificaciones	
Descripción: Verificar que funcione el sistema de notificaciones en el software.	
Condiciones de ejecución: Ya sea usuario o administrador tiene que estar logeado en el sistema	
Entradas: <ol style="list-style-type: none"> 1. Cuando el usuario o administrador se logea le van a llegar las notificaciones de las publicaciones que aún no han visto 2. Ejecutar el botón de refrescar 	
Resultado esperado: El sistema de notificaciones funciona ya que al logearse un usuario o administrador les va a mostrar aquellas publicaciones que aún no ha visto. Al ejecutar el botón de refrescar se actualizan todas aquellas notificaciones no vistas. (ver anexos 7)	
Evaluación: Prueba satisfactoria	

Fuente: Elaboración propia.

3.3 Estudio de factibilidad

Al desarrollarse un software es preciso saber si será factible o no su producción, por lo que hay que tener en cuenta los costos y beneficios que traerá consigo. Es necesario también realizar una estimación del esfuerzo, el tiempo de desarrollo y la cantidad de

personas que participarán para poder determinar eficazmente si resulta beneficioso el desarrollo, aunque es importante señalar que solo se habla de una estimación.

Ya que se emplea XP como metodología para el desarrollo del software, se tiene un estimado del tiempo de desarrollo del mismo, aprovechando que la fórmula de Bohem comprende este parámetro ($\text{Costo} = \text{Cantidad de Hombres} * \text{Salario Medio} * \text{Tiempo de Desarrollo}$), es posible obtener una estimación del salario del autor. Haciendo los cálculos pertinentes tenemos, considerando como valor promedio para el salario mensual por hombres \$ 2000.00, un desarrollador y un tiempo de desarrollo estimado de 5 meses, al sustituir y calcular se obtiene un costo de \$10000.00 pesos.

3.4 Análisis de los resultados obtenidos

Después de desarrollar todo un proceso de pruebas se lograron resultados satisfactorios, pues tras la detección de diferentes errores, obtenidos fundamentalmente con las pruebas funcionales, se solucionaron varios problemas que impedían el cumplimiento de los requisitos fundamentales del sistema.

Como resultados finales se obtuvo una aplicación web con una apariencia agradable y fácil de usar. La planificación inicial se cumplió en su totalidad, se utilizaron herramientas actuales para su desarrollo y el plan de entrega fue cumplido con éxito.

Se obtuvo un sistema en el que se pueden publicar noticias, archivos adjuntos, enlaces externos, logrando así una mejor comunicación y organización entre los departamentos del Parque Científico Tecnológico.

Se conoce las opiniones de los visitantes que llegan al PCT y la fecha de su visita.

El cliente quedó complacido con el trabajo y se espera que se logren alcanzar mayores resultados.

3.5 Conclusiones de capítulo

En este capítulo se llegó a la conclusión de que la planificación fue acertada, pues permitió el desarrollo del sistema según el cronograma y cumplimiento con los objetivos del cliente al iniciar esta investigación.

realización de pruebas de aceptación durante el desarrollo del sistema permitió tanto al cliente como al desarrollador encontrar defectos y errores en el sistema y su rápida corrección.

A partir del diseño de la propuesta y con el uso de las tecnologías adecuadas para su implementación se obtuvo un sistema que cumple a cabalidad con las expectativas del cliente.

Conclusiones

Una vez desarrollado el presente trabajo se arriba a lo siguiente:

1. EL análisis de los antecedentes y el flujo actual de los procesos relacionados con el objeto de estudio, permitió adquirir la información necesaria para dar solución a la problemática planteada.
2. A partir de la utilización de las etapas de la metodología de desarrollo de software XP para la elaboración de la aplicación se logró una comunicación directa y amplia con el cliente y buena respuesta ante los cambios sin importar la etapa del ciclo de vida del proyecto.
3. Se diseñó una aplicación web que facilita y controla adecuadamente el proceso de publicaciones de los trabajadores entre distintos departamentos, así como un registro de visitantes.
4. Con el desarrollo de las pruebas funcionales y el análisis de los resultados obtenidos, se permite verificar y demostrar el correcto funcionamiento de esta aplicación.

Recomendaciones

Teniendo en cuenta los resultados obtenidos y la continuidad del trabajo se recomienda:

- Incluir en la aplicación web para futuras versiones otras funcionalidades para extender su alcance, por ejemplo, la divulgación de aplicaciones desarrolladas por el PCT.
- En el registro de visitantes se propone insertar la firma digital como constancia de su visita.

Bibliografías

Acibeiro, M. (2022). *Qué es Node.js y para qué sirve.* Blog de LucusHost. <https://www.lucushost.com/blog/que-es-node-js/>

Aguilar, J. (2014, 13 julio). *Navicat MySQL.* Jose Aguilar Blog. <https://www.jose-aguilar.com/blog/navicat-mysql/>

Arjonilla R. (2019). *Qué es el Backend de una web y por qué es tan importante.* <http://rafarjonilla.com/que-es/backend>

Blog, R. (2018, 29 noviembre). *Requerimientos Funcionales y No Funcionales, ejemplos y tips.* Medium. <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>

Calero, S.M., (2003). Una explicación de la programación extrema (XP). V Encuentro usuarios xBase. MADRID.

Calvo, L. (2022). *¿Qué es una intranet corporativa y cómo funciona?* Blog. <https://es.godaddy.com/blog/que-es-intranet-corporativa/>

Cascante, P. & Fonseca, M. (2006). *Intranet: Una ventana a la comunicación interna de las organizaciones.* Revista de Derecho y Tecnologías de la Información. <http://www.uned.ac.cr/redti/cuarta/art2.pdf>

Claburn, T. (2007, 25 abril). *Google Releases Improved MySQL Code.* InformationWeek. <http://www.informationweek.com/government/google-releases-improved-mysql-code>

Clark, S. (2021, 28 abril). *New Mobile Apps Using HTML5, CSS & JS.* HTML Goodies. <https://www.htmlgoodies.com/webmaster/html5-css-js-mobile-apps>

COR Academy. (2022, 2 noviembre). *¿Next.js es lo Mismo que React.js?* COR. <https://projectcor.com/es/blog/next-js-es-lo-mismo-que-react-js/>

Cristancho, F. (2022). *¿Qué es un framework en programación? - Talently.* Talently Blog. <https://talently.tech/blog/que-es-un-framework-en-programacion/>

DB-Engines Ranking. (2022, 17 noviembre). DB-Engines. <https://db-engines.com/en/ranking>

Digite. (2021, 25 agosto). *Historias De Usuarios: Qué Son Y Por Qué Y Cómo Usarlas.*
<https://www.digite.com/es/agile/historias-de-usuarios>

Domínguez, M. (2005, septiembre). Todo Programación. Nº 12. Págs. 48-51. Editorial Iberprensa (Madrid). *Bases de datos en el cliente con JavaScript DB*

ECMA-262. (2022, 15 julio). Ecma International. <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>

elFaro Studio. (2022, 9 febrero). *Next.js: ¿qué es y por qué debes usarlo?* TodoXampp. <https://todoxampp.com/next-js-que-es-y-por-que-debes-usarlo/>

Escat Cortés, M. (2007, 12 enero). Diferencia entre comunicación e información. Disponible en: <http://www.gestiopolis.com/canales/gerencial/articulos/43/difcominf.htm>.

Fandom. (2014, 22 septiembre). *¿Recomendaría el uso de otra metodología | Wiki Metodología-xp . . .?* Wiki Metodología-xp.

García, I. J. B. (2021). *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?* <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>

Gonzales Ugarte, JL. (2003). Gestión del conocimiento en intranets corporativas: desarrollo de un modelo de diseño. Ibersid. Disponible en: <http://www.etse.urv.es/bibweb/jlgu/presentacions/saragossa.ppt/>

Gonzales, E. (2006). ¿QUÉ ES Y PARA QUÉ SIRVE HTML? EL LENGUAJE MÁS IMPORTANTE PARA CREAR PÁGINAS WEB. https://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=435:ique-es-y-para-que-sirve-html-el-lenguaje-mas-importante-para-crear-paginas-webs-html-tags-cu00704b&catid=69&Itemid=192

Halcomb, E., Hickman, L. (2015). Mixed Methods Research

KeepCoding, R. (2022). *Por qué usar JavaScript: ventajas y desventajas.* KeepCoding Tech School. <https://keepcoding.io/blog/porque-usar-javascript>

Laravel. (2022). *The PHP Framework for Web Artisans.* <https://laravel.com/>

Lardinois, F. (2015, 29 abril). [«Microsoft Launches Visual Studio Code, A Free Cross-Platform Code Editor For OS X, Linux And Windows»](#)

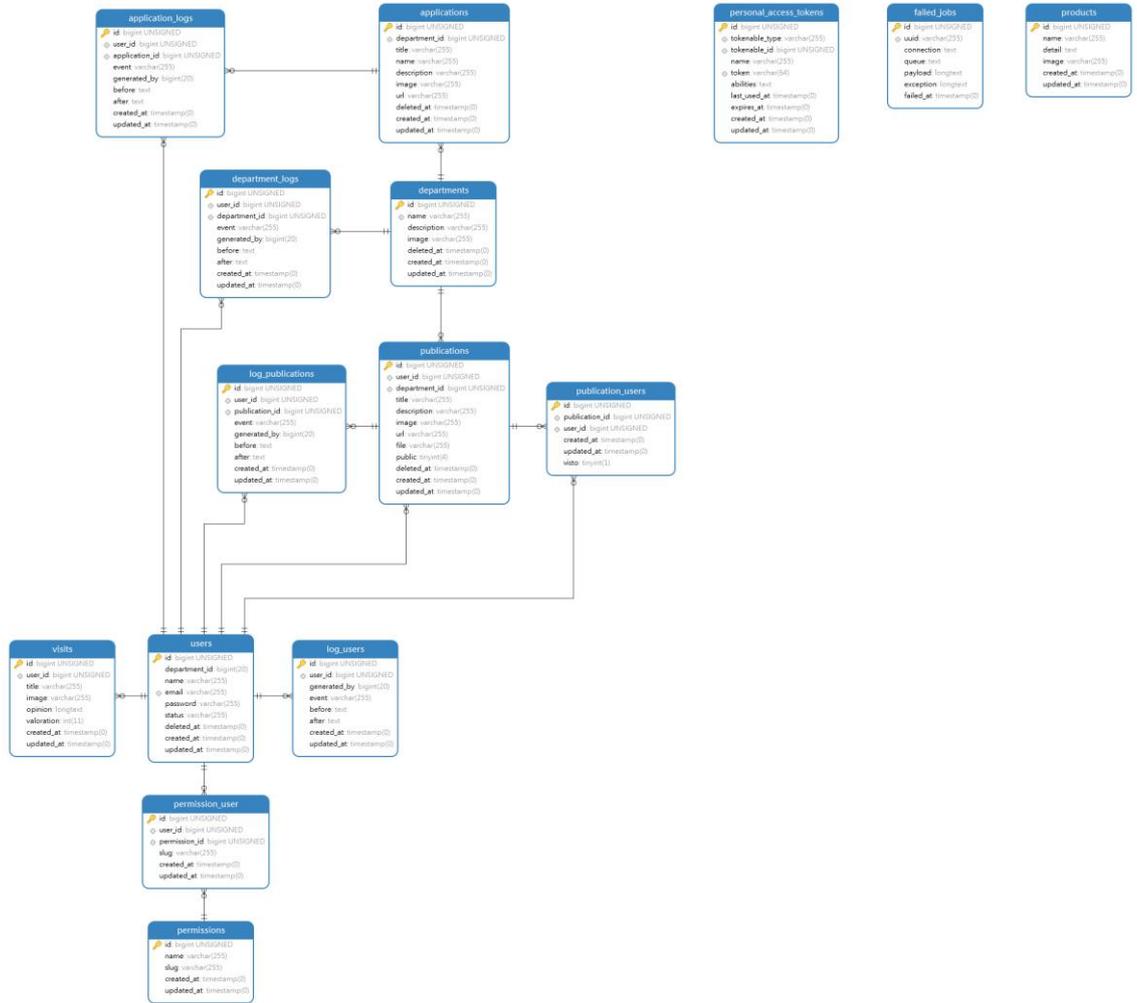
Letelier, P.& P., M^aCarmen, (2003). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).

Liferay. (2022, 19 agosto). *Cinco ejemplos de Intranet que potencian la productividad.* <https://www.liferay.com/es/blog/employee-experience/cinco-ejemplos-de-intranet-que-potencian-la-productividad>

- Mehdi Achour, Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., Seguy, D. & Vrana, J. (2022). *PHP: PHP Manual* - Manual. <https://www.php.net/manual/en/index.php>
- Monster Digital Agency. (2022). *Framework: Qué es y Todas sus Ventajas*. Epitech España. <https://www.epitech-it.es/que-es-un-framework/>
- Monster Digital Agency. (2022, febrero 10). *Qué es PHP y para qué sirve?* Epitech España. <https://www.epitech-it.es/que-es-php>
- Payne, R. & Corrales, E. (2022). *HTML Goodies: HTML, CSS & JavaScript Development Resources*. HTML Goodies. <https://www.htmlgoodies.com/>
- Portal, T. (2022, 1 septiembre). *Intranet*. TIC Portal. <https://www.ticportal.es/glosario-tic/intranet>
- Pressman, Roger S. (2015). "Ingeniería del Software: Un Enfoque Práctico". Quinta Edición
- Pujol, ME. (2007, 3 marzo). Internet, Intranets, Extranets ¿Son importantes en la empresa periódica?. Disponible en: <http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/eulalia.htm>
- Romero, L.D.J. (s.f.). Una Introducción a la Programación Extrema
- Sharma, P. (2022). *Las 10 principales ventajas de los servicios de desarrollo de Laravel para empresas*. Cynoteck. https://cynoteck.com/wp-content/plugins/gtranslate/url_addon/gtranslate.php?glang=es&qurl=blog-post/top-10-advantages-of-laravel-development-services-for-enterprises
- Telot, D.J.A. (2010). Soporte Conceptual. Selección y elaboración de los Métodos de Investigación in METODOLOGÍA DE LA INVESTIGACIÓN CIENTÍFICA. Facultad de Informática UMCC.
- Turrado, J. (2020, 10 marzo). *Qué son las pruebas de software*. campusMVP.es. <https://www.campusmvp.es/recursos/post/que-son-las-pruebas-de-software.aspx>
- UMCC. (2022). *Parque Científico Tecnológico de Matanzas – Universidad de Matanzas*. <http://www.umcc.cu/parque-cientifico-tecnologico-de-matanzas/>
- Zúñiga, F. G. (2022). *¿Qué es Laragon y diferencias con Xampp?* Blog de arsys.es. <https://www.arsys.es/blog/introduccion-a-laragon>

Anexos

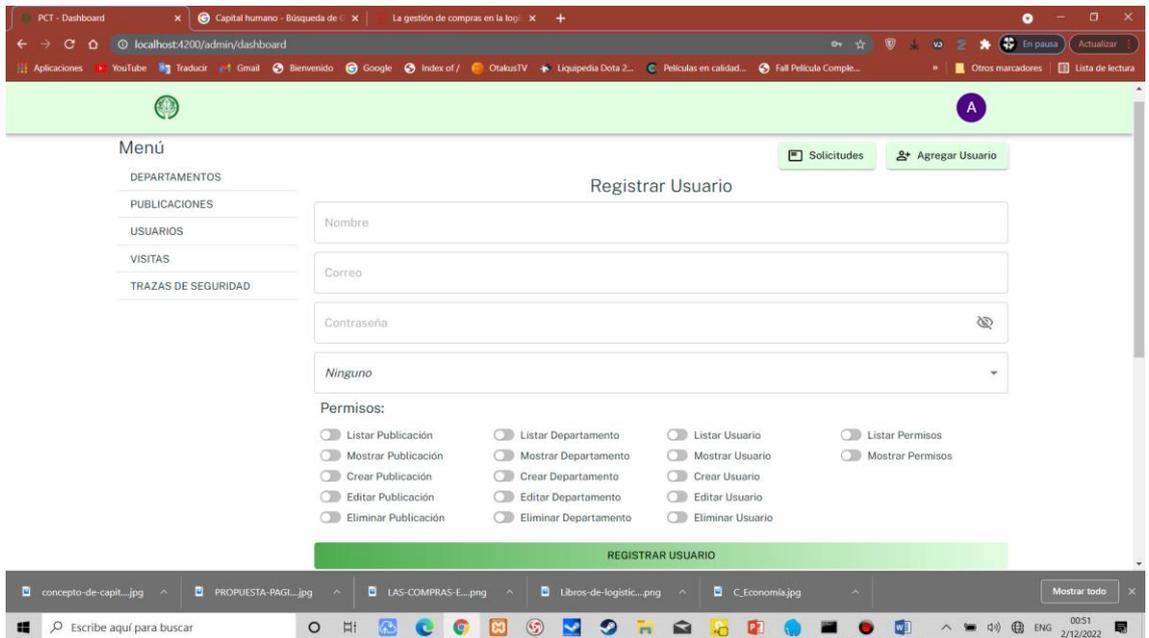
1.

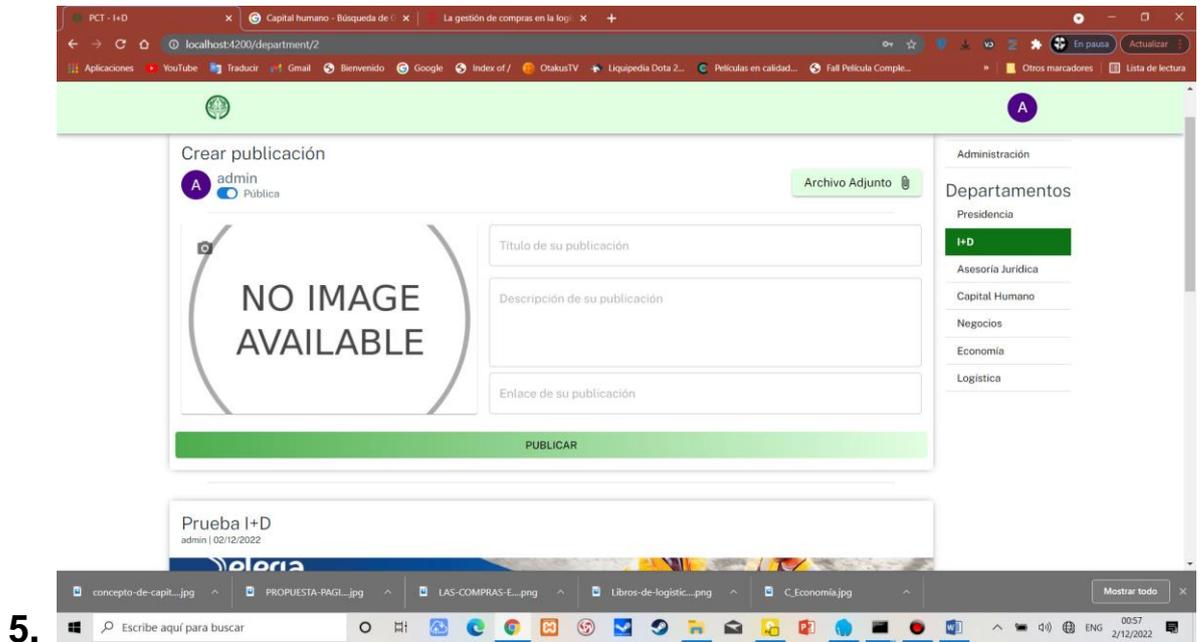
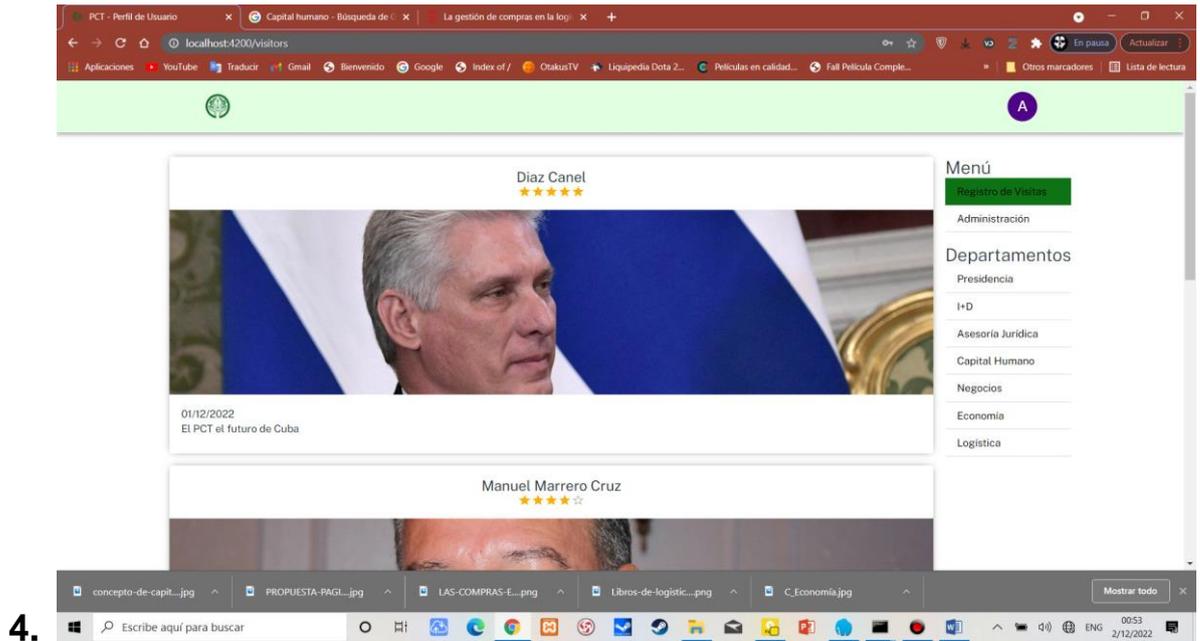


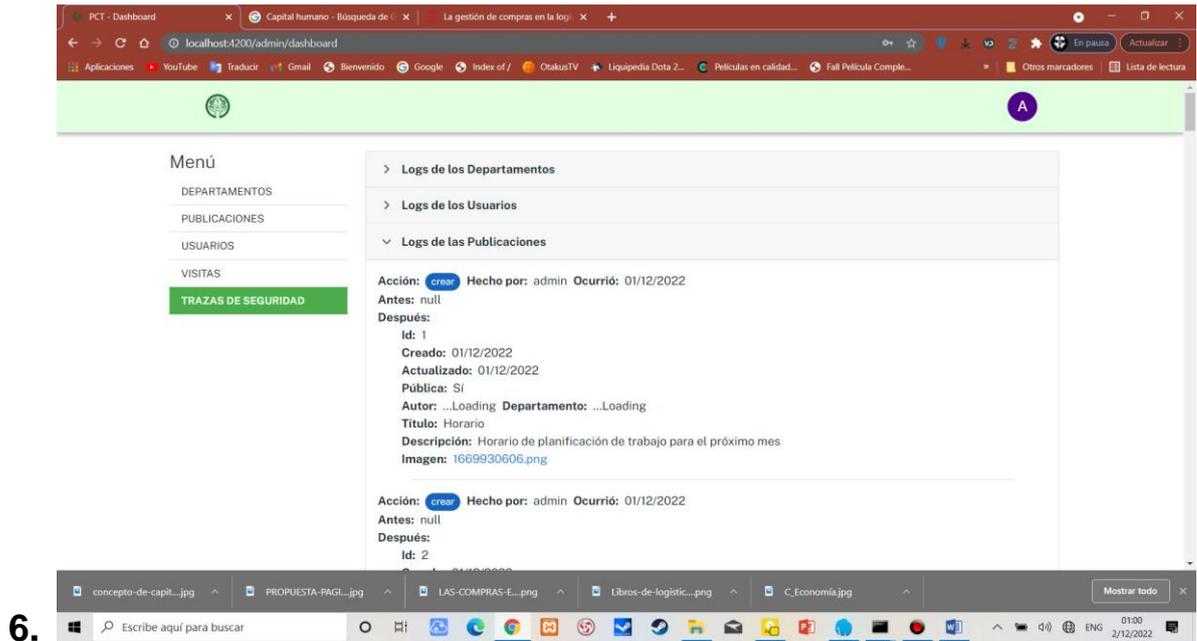
2.



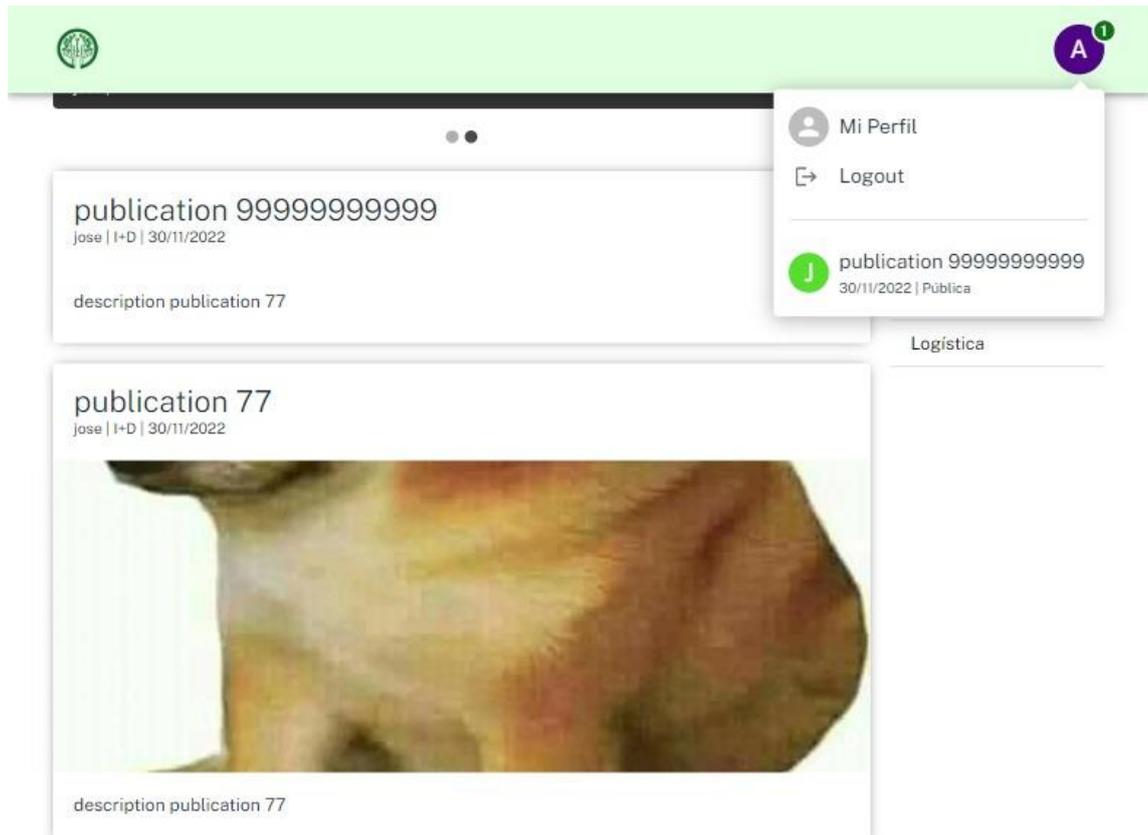
3.







6.



7.