

**Aplicación web de gestión bibliotecaria para promover y analizar la actividad lectora en
las PPL (Persona Privada de la Libertad) del EPAMS Girón Santander**

Dannys Eduardo Cardozo Benitez

Asesor.

Martin Camilo Cancelado Ruiz

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI

Programa de Ingeniería de Sistemas

2022

Dedicatoria

Quiero agradecer a toda mi familia, a mi madre, mi hermana y mis hijos por el apoyo incondicional que siempre me han brindado en mis metas a pesar de la distancia, siempre sentí ese apoyo y respaldo en la distancia, sin su apoyo esto no hubiera sido posible porque muchas veces pensé en renunciar a esto y ustedes me alentaron a seguir adelante. Gracias los quiero con todo mi corazón.

Agradecimiento

Le agradezco muy profundamente a mis tutores por su dedicación y paciencia, por ese apoyo y palabras de aliento, sin sus correcciones no hubiese sido posible llegar a esta instancia. Gracias por sus consejos, les agradeceré por siempre en mi desempeño como profesional.

Resumen

Con el desarrollo del proyecto se pretende elaborar una herramienta de información y administración mediante la cual se facilitará el acceso al material bibliográfico de la biblioteca Alejandría y mejorará la administración de los datos. El aplicativo está pensado para funcionar en ambiente web, mejorando el acceso de las Personas Privadas de la Libertad (PPL) al consultar el material bibliográfico y de esta manera incentivar la actividad lectora, también mejorara la administración de los datos y facilitara su análisis.

Para su funcionamiento el aplicativo tendrá tres usuarios fundamentales, interno, bibliotecario y administrador, y mediante una interfaz de usuario amigable los internos podrán consultar el material bibliográfico de su preferencia y solicitar su préstamo, con esta operación el bibliotecario podrá generar préstamos y analizar dicha información.

El aplicativo además de ser un instrumento para mejorar el acceso al material bibliográfico; recolectará tendencias de uso e información que servirá de soporte para la toma de decisiones que optimicen el proceso resocializador en los centros penitenciarios de Colombia.

La construcción de esta herramienta se ejecuta en cinco fases principales que son: 1) La caracterización del problema, 2) Análisis y Pre-diseño, 3) Diseño e implementación de la versión de prueba de la herramienta propuesta, 4) Prueba piloto, 5) Documentación de resultados, análisis y conclusiones del modelo planteado.

Palabras clave: análisis, diseño, programación, aplicativo, administración.

Abstract

With the development of this project, the aim is to create an information and management tool that will facilitate access to the bibliographic material of the Alejandría Library and improve data management. The application is designed to operate in a web environment, enhancing access for Persons Deprived of Liberty (PDL) to consult bibliographic material and, in this way, encourage reading activity. It will also improve data management and facilitate its analysis.

For its operation, the application will have three main types of users: inmate, librarian, and administrator. Through a user-friendly interface, inmates will be able to browse the bibliographic material of their choice and request its loan; with this operation, the librarian will be able to process loans and analyze the corresponding information.

In addition to being a tool to improve access to bibliographic material, the application will collect usage trends and information that will serve as support for decision-making aimed at optimizing the resocialization process in Colombia's penitentiary centers.

The development of this tool will be carried out in five main phases: 1) Problem characterization, 2) Analysis and preliminary design, 3) Design and implementation of the proposed trial version of the tool, 4) Pilot test, 5) Documentation of results, analysis, and conclusions of the proposed model.

Palabras clave: analysis, desing, programming, application, administration.

Tabla de Contenido

Introducción	15
Planteamiento del problema	15
Justificación	20
Objetivos	22
Objetivo general	22
Objetivos Específicos	22
Marco de Referencia	23
Marco Teórico	23
<i>Evolución del software</i>	23
Ingeniería de Software	26
<i>Fases de Desarrollo en la Ingeniería de Software</i>	28
<i>Modelos para el Diseño de Software</i>	29
Modelos y Metodologías	30
Los distintos modelos y metodologías para el desarrollo de software son:	30
<i>Modelo en Cascada</i>	30
<i>Modelo de Prototipos</i>	30
<i>Modelo en Espiral</i>	31
<i>Desarrollo por Etapas</i>	31
<i>Desarrollo Iterativo y Creciente</i>	32
<i>RAD o Desarrollo de Aplicaciones Rápidas</i>	32
<i>Desarrollo Concurrente</i>	33
<i>Proceso unificado</i>	33
<i>RUP o proceso unificado relacional</i>	33
Metodologías de desarrollo de Software	34
Metodologías Ágiles Contra Metodologías Tradicionales	36
<i>Metodologías Tradicionales</i>	36
<i>Metodologías Ágiles</i>	36
Comparación de las Metodologías	37
Paradigmas en el Desarrollo de Software	39
<i>Paradigma imperativo, también llamado paradigma por procedimientos</i>	39
<i>Paradigma funcional</i>	39
<i>Paradigma declarativo, también conocido como paradigma de programación lógica</i>	39
<i>Paradigma reactivo</i>	40

<i>Paradigma orientado a objetos</i>	40
Boceto del Software	40
<i>Investiga a tu usuario y el diseño de (User Experience) UX</i>	41
<i>Define el diseño de tu página web</i>	41
<i>Diseño de información</i>	42
Inclusión de Interfaces Gráficas en el Software.....	42
Marco Conceptual.....	44
Ingeniería de requerimientos.....	44
El hardware	48
El software.....	49
Las bases de datos	49
Un Sistema de Gestión de Bases de Datos (SGBD).....	49
<i>Evitar la redundancia de datos</i>	51
<i>Consistencia de datos</i>	51
<i>Almacenamiento</i>	51
<i>Accesibilidad</i>	51
<i>Centralización</i>	52
<i>Integridad de los datos</i>	52
<i>Seguridad</i>	52
Ventajas de un Sistema de Gestión de Base de Datos.....	52
<i>Organización</i>	52
Desventajas de un Sistema de Gestión de Base de Datos	53
<i>Costos operacionales</i>	53
<i>Vulnerabilidad</i>	53
Web	54
Marco Contextual.....	56
Delimitaciones	62
Delimitación de la Metodología de Desarrollo de Software	62
Las Metodologías de Desarrollo De Software	63
<i>Aplicación al Proyecto</i>	63
Arquitectura de Software Modelo Vista Controlador.....	68
El Modelo.....	68
La Vista	68
El Controlador.....	68

El modelo es el responsable	68
El controlador es responsable.....	69
Las vistas son responsables de	69
Patrones de Diseño de Software.....	70
<i>¿Qué es un Patrón de diseño?</i>	70
Herramientas	72
<i>Framework para Desarrollo de Aplicaciones Web Laravel</i>	72
<i>¿Por qué utilizar laravel en el proyecto?</i>	72
MySQL.....	73
<i>Compatible</i>	73
<i>Actualizable</i>	73
<i>Costo</i>	73
XAMPP	74
¿Pero, Que Es XAMPP?	74
<i>Linux</i>	75
<i>Apache</i>	75
<i>MySQL/MariaDB</i>	75
<i>PHP</i>	75
<i>Perl</i>	75
Visual Studio Code.....	76
<i>Características</i>	76
Tecnologías utilizadas en el desarrollo del proyecto	77
<i>HTML</i>	77
<i>CSS</i>	78
<i>PHP</i>	78
Modelo Cliente-Servidor.....	79
<i>Java Script</i>	80
Base De Datos	81
<i>SQL</i>	81
Seguridad Informática	82
Lenguaje Unificado De Modelado(UML) e Ingeniería De Software	83
<i>¿Porque Utilizar UML en el Proyecto?</i>	84
Diseño y Desarrollo Del Proyecto	85
Modelo De Desarrollo.....	85

<i>Metodología SCRUM</i>	85
¿Qué es Scrum?.....	86
<i>Roles Scrum</i>	86
<i>Elementos Scrum</i>	87
Dinámica	88
<i>Iteraciones</i>	88
<i>Reunión de planificación de Sprint</i>	89
Alcance	90
Alcance Del Producto.....	90
Alcance del proyecto.....	90
Personal Involucrado	91
Referencias.....	92
Funcionalidad del producto.....	93
Requerimiento general para PreLibrerate	93
Características de los Usuarios.....	97
Restricciones	97
Suposiciones y dependencias	98
Evolución previsible del sistema.....	98
Requisitos Específicos	99
Requisitos comunes de las interfaces	99
<i>Interfaces de usuario</i>	99
<i>Cabezotes</i>	99
<i>Pie de Paginas</i>	100
Interfaces de hardware	101
Interfaces de software.....	101
Interfaces de comunicación.....	101
Requerimientos Funcionales	102
Diseño.....	105
Diagrama de Casos de Uso.....	108
<i>Identificación de Actores</i>	108
<i>Casos de uso del Usuario</i>	109
<i>Casos de Uso del Usuario Registrado</i>	109
<i>Casos de uso del PPL</i>	110
Casos de uso Bibliotecario	110

<i>Casos de uso del administrador</i>	111
Diagramas de Actividades.....	112
Diagrama de Clases	115
Diagrama de Secuencia	115
Implementación.....	116
Diseño de la Interfaz de Usuario	116
Software Versión Beta	121
Dirección	129
<i>Descripción de las fases de entrega</i>	129
Delimitación Temporal.	130
Cronograma de Desarrollo del Proyecto.....	130
Conclusion	132
Referencias Bibliográficas	133

Lista de Tablas

Tabla 1 <i>Comparación Metodológica</i>	38
Tabla 2 <i>Selección de Metodología Por Cada Aspecto de la Comparación</i>	63
Tabla 3 <i>Ventajas y Desventajas de MySQL</i>	74
Tabla 4 <i>Personal Involucrado</i>	91
Tabla 5 <i>Definiciones, Acrónimos y Abreviaturas</i>	91
Tabla 6 <i>Referencias</i>	92
Tabla 7 <i>Usuario Administrador</i>	97
Tabla 8 <i>Usuario Bibliotecario</i>	97
Tabla 9 <i>Usuario PPL</i>	97
Tabla 10 <i>Abreviaturas Para Requerimientos Funcionales</i>	102
Tabla 11 <i>Requerimientos Funcionales Para Almacenamiento</i>	103
Tabla 12 <i>Requerimientos Funcionales Para Informes y Reportes</i>	104
Tabla 13 <i>Requerimientos Funcionales Para Procesos</i>	105
Tabla 14 <i>Cronograma de Desarrollo del Proyecto</i>	130
Tabla 15 <i>Recursos Necesarios</i>	131

Lista de figuras

Figura 1 <i>Flujo Del Modelo Vista Controlador</i>	69
Figura 2 <i>Patron Modelo Vista Controlador</i>	71
Figura 3 <i>Modelo Cliente Servidor</i>	79
Figura 4 <i>Vista de la Consulta Libro más Leído en MySQL</i>	93
Figura 5 <i>Vista del Resultado de la Consulta Libro Mas Leído en MySQL</i>	94
Figura 6 <i>Vista en el Front-End de la Consulta Libro más Leído</i>	94
Figura 7 <i>Vista Listado de Lectores en MySQL</i>	95
Figura 8 <i>Vista del Resultado de la consulta Listado Lectores en MySQL</i>	95
Figura 9 <i>Boceto del Front-End en la sesión lector destacado</i>	96
Figura 10 <i>Vista en el Front-End en el Navegador en la Sesión Lector Destacado</i>	96
Figura 11 <i>Alternativa 1 de Encabezado Diseño Herramienta Balsamiq</i>	99
Figura 12 <i>Alternativa 2 de Encabezado en Herramienta Balsamiq</i>	99
Figura 13 <i>Alternativa de Encabezado en el Navegador</i>	100
Figura 14 <i>Alternativa 1 Pie de Página, Diseño en Herramienta Balsamiq</i>	100
Figura 15 <i>Alternativa 2 Pie de Página en Herramienta Balsamiq</i>	100
Figura 16 <i>Alternativa 2 Pie de Página en Navegador</i>	101
Figura 17 <i>Modelo Lógico de la Base de Datos</i>	106
Figura 18 <i>Modelo Relacional de la Base de Datos</i>	107
Figura 19 <i>Diagrama Caso de Uso Identificación de Actores</i>	108
Figura 20 <i>Diagrama Caso de Uso Usuarios</i>	109
Figura 21 <i>Diagrama Caso de Uso Usuario Registrado</i>	109
Figura 22 <i>Diagrama Caso de Uso Usuario PPL</i>	110

Figura 23 <i>Diagrama Caso de Uso Usuario Bibliotecario</i>	110
Figura 24 <i>Diagrama Caso de Uso Administrador</i>	111
Figura 25 <i>Diagrama de Actividad</i>	112
Figura 26 <i>Diagrama de Actividad Gestión de Usuarios</i>	113
Figura 27 <i>Diagrama de Actividad. Insertar, Actualizar, Eliminar y Generar Préstamo</i>	114
Figura 28 <i>Diagrama de Clases</i>	115
Figura 29 <i>Diagrama de Secuencia</i>	115
Figura 30 <i>Modulo de Registro Usuario</i>	116
Figura 31 <i>Vista Inicio de Sesión en el Navegador</i>	117
Figura 32 <i>Vista Administración de Biblioteca en el Navegador</i>	117
Figura 33 <i>Diseño Registro de Libro en Herramienta Balsamiq</i>	117
Figura 34 <i>Vista Registro de Libro en Herramienta Balsamiq</i>	118
Figura 35 <i>Diseño de Eliminación de Libro en Herramienta Balsamiq</i>	118
Figura 36 <i>Diseño de Actualización de Libro en Herramienta Balsamiq</i>	118
Figura 37 <i>Vista de Actualización de Libro en Navegador</i>	119
Figura 38 <i>Diseño de Catálogo de Libro en Herramienta Balsamiq</i>	119
Figura 39 <i>Vista de Más sobre Nosotros en navegador</i>	119
Figura 40 <i>Vista Completa de la Interfaz de Usuario en Herramienta Balsamiq</i>	120
Figura 41 <i>Vista del Front- End Completa en Navegador</i>	120
Figura 42 <i>Vista Desde Varios Tamaños de Dispositivos Página Header</i>	121
Figura 43 <i>Vista Desde Varios Tamaños de Dispositivos, Página Header</i>	122
Figura 44 <i>Vista Desde Varios Tamaños de Dispositivos, Páginas Eventos y Actividades</i>	122

Figura 45 <i>Vista Desde Varios Tamaños de Dispositivos, Páginas Géneros Literarios Más Leídos</i>	123
Figura 46 <i>Vista Desde Varios Tamaños de Dispositivos, Página Blog</i>	123
Figura 47 <i>Vista en Navegador, Página Nosotros</i>	124
Figura 48 <i>Vista en Navegador, Página Anuncios</i>	124
Figura 49 <i>Vista en Navegador, Página Blog</i>	125
Figura 50 <i>Vista en Navegador, Página Inicio de Sesión</i>	125
Figura 51 <i>Vista en Navegador del Back-End, Administración</i>	126
Figura 52 <i>Vista en Navegador del Back-End, Crear Libro y Alertas de Navegación</i>	126
Figura 53 <i>Vista en Navegador del Back-End, Crear Libro</i>	127
Figura 54 <i>Vista en Navegador del Back-End, Actualización de Libro</i>	127
Figura 55 <i>Vista del Modelo Relacional en MySQL</i>	128
Figura 56 <i>Vista de phpMyAdmin XAMPP</i>	128

Introducción

Planteamiento del problema

La siguiente investigación tiene como campo de aplicación las bibliotecas penitenciarias de Colombia, especialmente la biblioteca Alejandría del Establecimiento Penitenciario de Alta y Mediana Seguridad (EPAMS) Girón en el departamento de Santander. Con el desarrollo de este proyecto se pretende mejorar el servicio bibliotecario en cuanto a atención al usuario y administración. La biblioteca Alejandría funciona dentro de las instalaciones del Instituto Integrado Enrique Low Murtra (IIELM) del EPAMS Girón en el departamento de Santander y tiene como objetivo primordial brindar apoyo a la Institución Educativa suministrando el material bibliográfico a la planta docente y alumnado en general; aparte de esto suministrar el material bibliográfico a 10 bibliotecas articuladas con la biblioteca central (biblioteca Alejandría), estas funcionan en los diez pabellones del penal y su misión es facilitar el acceso al material bibliográfico a los más de 1800 internos, del total de esta población solo un 25% utiliza los recursos bibliotecarios; ya que acceder ellos tiene muchas trabas pues no hay una política destinada por el INPEC (Instituto Nacional Penitenciario y Carcelario) para incentivar la lectura; los internos no tienen acceso directo a los ejemplares lo cual ha generado un inconformismo por parte de la población reclusa pues no pueden consultar de manera directa la sinopsis del libro, solo tienen acceso a un catálogo, este relaciona el título y el autor del libro lo cual desmotiva a los internos al momento de decidir que leer.

La ley 65 de 1993 ordena en su artículo 94. Título VIII EDUCACION Y ENSEÑANAZA lo siguiente: "...En las penitenciarias, colonias y cárceles de distrito judicial, se organizarán sendas bibliotecas. Igualmente, en el resto de los centros de reclusión se promoverá y estimulará entre los internos, por los medios más indicados, el ejercicio de la lectura". De acuerdo con los datos proporcionados por la división Social del INPEC, hay treinta (30)

establecimientos que cuentan con centros educativos conformados como tales (de acuerdo con las “pautas”).

Esto equivale a 21.58% de establecimientos con Centros Educativos. En lo relacionado con la existencia de bibliotecas hay aproximadamente cuarenta (40) establecimientos que cuentan con áreas destinadas al préstamo de libros y de lectura, aunque la mayoría de los establecimientos tienen, aunque sea de manera incipiente un esbozo de biblioteca, que adolecen de contar con ejemplares desactualizados, deteriorados o incompletos y los espacios son generalmente bastante reducidos (repasemos lo anotado acerca de la infraestructura carcelaria). Las bibliotecas casi siempre se nutren de las donaciones logradas por campañas de recolección de libros que llevan a cabo los funcionarios encargados de las áreas educativas orientadas a tocar la sensibilidad de los estudiantes de colegios o de particulares. La calidad de los textos es cuestionable, no son bibliotecas planeadas y su elemental organización se deriva de la existencia de textos que han sido prácticamente desechados en otros sitios.

Vale señalar que no existe en el presupuesto de la entidad un rubro destinado al mantenimiento de las bibliotecas, ni adquisición de libros, ni dotación de equipos básicos para el funcionamiento de estas. Las bibliotecas que existen funcionan en las áreas educativas y son de uso casi exclusivo de los estudiantes, pues la falta de disponibilidad de guardia hace imposible que los internos en los patios (los que no estudian ni trabajan) se desplacen hasta el área educativa. La cárcel no es un espacio donde se movilicen las personas a su voluntad, y el solo desplazamiento hacia la biblioteca puede significar traspasar tres, cuatro, cinco, seis o más rejas y ello significaría contar con una unidad de guardia que acompañe al solicitante, lo cual no es posible por el déficit de unidades de guardia. Los internos en los patios son víctimas del ocio o en el mejor de los casos participan solamente en programas de educación informal (talleres,

conferencias, etc.) o programas eventuales de prevención de drogadicción, asistencia espiritual, cine-foros y otros eventuales con los cuales no siempre adquiere un descuento de pena.

Generalmente, quien desee obtener el préstamo de un libro y no tenga acceso al área educativa, se debe valer del favor de otra persona que lo haga por él; de todas maneras, la sola actividad de lectura no le reporta el beneficio de la redención (que es lo más ansiado por la población carcelaria). En la Metodología Plan de Acción y Sistema de Oportunidades (P.A.S.O.) para alcanzar la resocialización a través del trabajo, estudio y enseñanza, actividad cultural, recreativa, deportiva y relaciones familiares que se está implementando a nivel nacional se establece el “programa para bibliotecarios: dirigido a internos con bachillerato completo o con un nivel académico elevado, interesados en liderar procesos de lectura como bibliotecarios. Se eligen internos que se encuentren preferiblemente fase de Alta seguridad, con el fin de que se puedan capacitar como líderes positivos.

El proceso para adquirir un libro lo hacen diligenciando un formato de préstamo el cual cada encargado de estas bibliotecas rodantes hacen llegar a la biblioteca central la cual genera los préstamos y los hace llegar por intermedio de estos bibliotecarios, con la información suministrada en estos formatos se alimenta la base de datos de la biblioteca, convirtiéndose en un trabajo tedioso ya que hay que registrar el historial de lectores para tener un control del movimiento de los libros y llevar una estadística con el fin de hacer un análisis de la actividad lectora del personal interno. Pero a pesar de tener un control sobre el inventario de la biblioteca el cual garantiza que el activo primordial de esta no se deteriore o pierda, lo ideal sería que todo el personal de internos pudiera tener acceso al catálogo y conocer la sinopsis del libro que desean; el acceso a la biblioteca es limitado teniendo en cuenta los protocolos de seguridad del penal, aparte de esto no se cuenta con un salón de lectura para satisfacer esta necesidad. La

biblioteca Alejandría cuenta con una aplicación la cual le permite tener control sobre el inventario y conocer el movimiento de los ejemplares; también permite conocer datos estadísticos como , porcentaje de lectores, la preferencias en cuanto a género literario, cual es el interno que más utiliza el servicio y otros datos relacionados con la actividad lectora de la población reclusa; este registro y control se ha visto afectado por el aumento de usuarios en la biblioteca, lo cual ha generado un represamiento de los registros y por ende un retraso en alimentar la base de datos, lo cual no permite tener un dato preciso de la actividad, dicha problemática ha creado la necesidad de buscar una solución la cual permita garantizar a la población reclusa la accesibilidad al material bibliográfico y se puedan tener datos en tiempo real de las actividades de la biblioteca y la actividad lectora en el centro de reclusión; teniendo en cuenta que los internos aunque no tienen acceso a la biblioteca pero si tienen acceso a la institución educativa la cual cuenta con varias salas de informática, estos espacios son ficha clave en la solución a dicha problemática.

En tanto, la problemática expuesta permite generar una alternativa de solución como es la creación de una aplicación en ambiente web con una interfaz amigable que permita a todos los PPL del penal que estudian en el IIELM tener acceso a Los catálogos de la biblioteca desde cualquiera de las salas de informática de la institución educativa, en ella después de registrarse podrán consultar el inventario de la biblioteca, leer la sinopsis del libro, hacer la solicitud del préstamo y comentar sobre la enseñanza del libro para su desarrollo personal de esta manera podremos tener un dato en tiempo real de la actividad lectora del interno, como son su preferencia en cuanto a género literario, solucionando el problema de accesibilidad y represamiento de información por falta de personal para ingresar esta; esta aplicación contara con una interfaz administrativa que recibirá en tiempo real las solicitudes de los usuarios y

generara los prestamos hacia los patios, de esta manera el administrador de la biblioteca y los usuarios podrán interactuar y se generara un ambiente amigable que hará que los internos se interesen a un más por la lectura.

Pregunta de investigación: En atención a la problemática expuesta anteriormente, se propone una alternativa de solución que implica la implementación de un software que permita a los internos de los penales de Colombia tener accesibilidad a las bibliotecas y se pueda conocer el comportamiento de dicha actividad en la población reclusa, de esta manera motivar e incentivar a los internos hacia la lectura convirtiéndola en una herramienta en su proceso resocializador. De allí, surge la pregunta:

¿Cómo dar accesibilidad al material bibliográfico, promover y analizar la actividad lectora de las PPL del EPAMS Girón Santander?

Justificación

Es un hecho que el gusto por la lectura se está perdiendo cada vez más en la actualidad. Es lamentable puesto que es indispensable para la educación de un pueblo y la creación de una conciencia crítica. Debido a la importancia de incentivar la lectura en la población reclusa del país, este proyecto busca que la actividad lectora en las cárceles del país sea de fácil acceso y crear espacios para que aumente el interés por la lectura.

Para poder mejorar la administración en la biblioteca Alejandría y mejorar la accesibilidad de las PPL del EPAMS Girón, la presente investigación propone desarrollar un software con un entorno amigable, de fácil acceso, comprensible, desde la cual los internos puedan consultar la base de datos de la biblioteca y reservar el libro de su preferencia, puedan hacer comentarios del servicio que se presta y hacer recomendaciones para mejorar dicho servicio, esto con el fin de generar interés en los PPL por la lectura y hacerlo sentir que hace parte del proceso resocializador y su opinión es importante para mejorar el servicio de biblioteca.

De la misma forma, se busca que al interactuar con el interno(usuario) este nos suministre la información necesaria para poder tener un registro sobre las preferencias bibliográficas de las PPL y poder hacer un análisis de su comportamiento lector, con esta información podremos conocer datos cuantificables de esta actividad como conocer el porcentaje de internos que utilizan en cierto periodo de tiempo la biblioteca , también se podrá suministrar esta información a la oficina de reinserción social y tratamiento y desarrollo del penal para la toma de decisiones en cuanto al proceso resocializador del interno pues esta área podrá conocer las preferencias bibliográficas del interno y las reflexiones que este haga de los libros consultados.

Con el desarrollo de este proyecto se busca fomentar una cultura lectora de las PPL y convertir la biblioteca Alejandría en un espacio en el cual los internos encuentren el material necesario para educarse, culturizarse y cambiar la manera de pensar, reflexionando sobre el

delito cometido y de esta manera prepararse para enfrentarse a una sociedad en la cual debe volver a encajar.

El desarrollo de este software convertirá la biblioteca Alejandría en una herramienta indispensable a la hora de hacer un análisis en el proceso resocializador por parte de la oficina de reinserción social del penal y en un espacio donde los internos encuentren las respuestas necesarias a sus necesidades.

Objetivos

Objetivo general

Implementar una herramienta de gestión bibliotecaria que contribuya a incentivar la lectura en los centros de reclusión, dando fácil acceso al material bibliográfico y registre el comportamiento lector de la población reclusa para una oportuna toma de decisiones en su proceso resocializador.

Objetivos Específicos

Facilitar el acceso al material bibliográfico de la biblioteca Alejandría a los internos del EPAMS Girón.

Registrar la actividad lectora de los internos del EPAMS Girón.

Incentivar la lectura en los centros de reclusión del país.

Establecer la eficiencia de la herramienta a través de un análisis de resultados.

Diseñar la versión de prueba de la herramienta propuesta de los módulos usuario (internos) y administrador.

Marco de Referencia

Marco Teórico

Evolución del software

El contexto en que se ha desarrollado el software está fuertemente ligado a las casi cinco décadas de evolución de los sistemas informáticos. Un mejor rendimiento del hardware, una reducción del tamaño y un coste más bajo, han dado lugar a sistemas informáticos más sofisticados.

A continuación, se describiré la evolución del Software dentro del contexto de las áreas de aplicación de los sistemas basados en computadoras.

Los primeros años (1950 - 1965)

- El software estaba en su infancia.
- El software era un añadido.
- Existían pocos métodos para la programación.
- No se tenía una planificación para el desarrollo del software.
- Los programadores trataban de hacer las cosas bien.
- El software se diseñaba a medida.
- El software era desarrollado y utilizado por la misma persona u organización

(entorno personalizado).

- El diseño de software era realizado en la mente de alguien y no existía documentación.

La segunda era (1965 - 1975)

- Multiprogramación y sistemas multiusuarios introducen nuevos conceptos de interacción hombre-máquina.

- Sistemas de tiempo real que podían recoger, analizar y transformar datos de múltiples fuentes.
- Avances en los dispositivos de almacenamiento en línea condujeron a la primera generación de sistemas de gestión de Base de Datos.
- Software como producto y la llegada de las "casas de software" produciéndose así una amplia distribución en el mercado.
- El software se desarrollaba para ser comercializado.
- Se empezó a distribuir software para grandes computadoras y minicomputadores.
- El mantenimiento de software comenzó a absorber recursos en una gran medida.
- Comenzó una crisis del software porque la naturaleza personalizada de los programas hizo imposible su mantenimiento.

Conforme crecía el número de sistemas informáticos, comenzaron a extenderse las bibliotecas de software de computadora. Las casas desarrollaban proyectos en que se producían programas de decenas de miles de sentencias fuente. Los productos de software comprados en el exterior incorporaban cientos de miles de nuevas sentencias. Una nube negra apareció en el horizonte. Todos estos programas tenían que ser corregidos cuando se detectaban fallos, modificados cuando cambiaban los requisitos de los usuarios o adaptados a nuevos dispositivos de hardware que se hubiera adquirido. Estas actividades se llamaron colectivamente mantenimiento del software.

La tercera era (1975 - 1985)

- Procesamiento Distribuido. Múltiples computadoras, cada una ejecutando funciones concurrentes y comunicándose con alguna otra.

- Redes de área local y de área global. Comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos.

- Amplio uso de microprocesadores y computadoras personales (hardware de bajo costo). Incorporación de "inteligencia" (autos, hornos de microondas, robots industriales y equipos de diagnóstico de suero sanguíneo). Impacto en el consumo.

- Planificación en el proceso del desarrollo de software.

La cuarta era (1985 -2000)

- Tecnología orientada a objetos.
- Los sistemas expertos y la inteligencia artificial se han trasladado del laboratorio a las aplicaciones prácticas.

- Software para redes neuronales artificiales (simulación de procesamiento de información al estilo de como lo hacen los humanos).

- Impacto colectivo del software.
- Sistemas operativos sofisticados, en redes globales y locales.
- Aplicaciones de software avanzadas.
- Entorno cliente/cliente servidor.
- Superautopista de información y una conexión del ciberespacio.
- La industria del software es la cuna de la economía.
- Técnicas de cuarta generación para el desarrollo de software.
- Programación de realidad virtual y sistemas multimedia.
- Algoritmos genéticos.
- Adopción de prácticas de Ingeniería del software.

Siglo XXI. Desde principios del siglo XXI se han dado grandes avances en todos los campos que abarca el software. Desde la aparición de los revolucionarios smartphone, con un software integrado que hace que un único dispositivo pueda realizar tareas en las que antes se necesitaba un ordenador. Por otra parte, también se han llegado a niveles de desarrollo en tecnologías aun por investigar y desarrollar como lo son la Inteligencia Artificial (IA) o las ya tan conocidas criptomonedas. Ejemplos de estos avances son los asistentes de voz de grandes compañías como Apple, Microsoft o Amazon, los cuales pueden imitar el comportamiento humano. A estos efectos, se dan multitud de lenguajes que se pueden utilizar y, por ende, que tipo profesional en específico se necesita. Algunos Lenguajes que podemos mencionar son SWIFT (para aplicaciones en entornos iOS), JavaScript (para entornos web) o PHP (Destinados a tareas y programación de servidores o maquinas físicas).

Ingeniería de Software

La Ingeniería de Software es una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería, y brindando soporte operacional y de mantenimiento. El campo de estudio de la ingeniería de software integra ciencias de la computación, ciencias aplicadas y las ciencias básicas en las cuales se encuentra apoyada la ingeniería.

Se citan las definiciones más reconocidas, formuladas por los siguientes prestigiosos autores:

Zelkovitz (1978) señala que “Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software”.

Bohem (1976) señala que “Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la

documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software”.

Bauer (1972) señala que “La ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales”.

La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación, y mantenimiento del software. *Standard Glossary of Software Engineering Terminology*.

En 2004, la U. S. Bureau of Labor Statistics (*Oficina de Estadísticas del Trabajo de Estados Unidos*) contó 760.840 ingenieros de software de computadora.

El término "ingeniero de software", sin embargo, se utiliza de manera genérica en el ambiente empresarial, y no todos los que se desempeñan en el puesto de ingeniero de software poseen realmente títulos de ingeniería de universidades reconocidas.

Algunos autores consideran que "desarrollo de software" es un término más apropiado que "ingeniería de software" (IS) para el proceso de crear software. Personas como Pete McBreen (autor de *Software Craftmanship*) cree que el término IS implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software.

Indistintamente se utilizan los términos "ingeniería de software" o "ingeniería del software"; aunque menos común también se suele referenciar como "ingeniería en software". En Hispanoamérica los términos más comúnmente usados son los dos primeros.

La creación del software es un proceso intrínsecamente creativo y la ingeniería del software trata de sistematizar este proceso con el fin de acotar el riesgo de fracaso en la

consecución del objetivo, por medio de diversas técnicas que se han demostrado adecuadas sobre la base de la experiencia previa.

La ingeniería de software se puede considerar como la ingeniería aplicada al software, esto es, por medios sistematizados y con herramientas preestablecidas, la aplicación de ellos de la manera más eficiente para la obtención de resultados óptimos; objetivos que siempre busca la ingeniería. No es solo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada.

La producción de software utiliza criterios y normas de la ingeniería de software, lo que permite transformarlo en un producto industrial usando bases de la ingeniería como métodos, técnicas y herramientas para desarrollar un producto innovador regido por metodologías y las buenas prácticas. Dicho producto es un medio que interviene en las funciones de sus usuarios para obtener un proceso productivo más eficaz y eficiente; hoy en día las empresas no podrían funcionar sin software porque este es un producto de uso masivo; por lo cual, el nivel de una empresa está determinado por la calidad de su infraestructura tecnológica y los productos desarrollados o adquiridos de acuerdo con sus necesidades.

Fases de Desarrollo en la Ingeniería de Software

La ingeniería de software es también conocida como desarrollo o producción de software y puede dividirse en tres partes fundamentales:

Definición. Esta fase, se centra en la concepción de la idea, se identifica que se requiere hacer y con qué finalidad, se define el comportamiento, las funcionalidades, entre otros aspectos, basándose en una serie de datos obtenidos, procesados y analizados en el levantamiento de la información.

Desarrollo. Esta fase, se centra en el diseño estructural de los datos, identificando el comportamiento y la manera en que se implementan las funciones dentro de la arquitectura del software.

Mantenimiento. Esta fase, se centra en el control y monitoreo del software diseñado, corrigiendo errores o realizando mejoras de acuerdo con las necesidades y requerimientos cambiantes por el contexto y el cliente.

Modelos para el Diseño de Software

Ingeniería de software es la aplicación de enfoques sistemáticos y disciplinados al desarrollo de software, para esto se han creado modelos y metodologías para la correcta utilización del tiempo y recursos que una empresa o entidad disponen.

A continuación, se listan los distintos modelos y metodologías del desarrollo de software:

- Modelo en cascada o Clásico (modelo tradicional).
- Modelo de prototipos.
- Modelo en espiral.
- Desarrollo por etapas.
- Desarrollo iterativo y creciente o Iterativo e Incremental.
- RAD.
- Desarrollo concurrente.
- Proceso Unificado.
- RUP.

Los modelos de desarrollo de software ofrecen un marco de trabajo usado para controlar el proceso de desarrollo de sistemas de información, estos marcos de trabajo consisten en una

filosofía de desarrollo de programas la cual debe de contar con las herramientas necesarias para la asistencia del proceso de desarrollo.

Modelos y Metodologías

Los distintos modelos y metodologías para el desarrollo de software son:

Modelo en Cascada

Este es el modelo en el cual se ordenan rigurosamente las etapas del desarrollo del software, de esto se obtiene que el inicio de una etapa de desarrollo deba de esperar el fin de la etapa anterior. De esto se obtiene que cualquier error detectado lleve al rediseño del área de código afectado, lo cual aumenta de costo el desarrollo del proyecto.

Modelo de Prototipos

Pertenece a los modelos evolutivos, en el cual el prototipo debe de ser construido rápidamente y con la utilización escasa de recursos. El prototipo es construido para mostrárselo al cliente, obtener críticas y retroalimentación, con lo cual se obtendrán los requisitos específicos para la aplicación a partir de las metas graficas que son mostradas. Las etapas de este modelo son:

- Plan rápido.
- Modelo.
- Construcción del prototipo.
- Entrega y retroalimentación.
- Comunicación.
- Entrega del desarrollo final.

Entre sus ventajas se encuentra que, es apto para el cliente que conoce a grandes rasgos el objetivo del software y a su vez, al equipo de desarrollo le ofrece una mejor visibilidad de la interacción del cliente con el software y el ambiente en el cual debe de realizarse.

Modelo en Espiral

Es el modelo en el cual las actividades se desarrollan en espiral, estas actividades se realizan conforme se van seleccionando de acuerdo con el análisis de riesgo. En cada iteración en este modelo, se deberán de tomar en cuenta los objetivos, las alternativas que se deberán de tomar de acuerdo con las características, estas son experiencia personal, requisitos a cumplir, las formas de gestión del sistema, entre otros. Este modelo tiene dos formas en las cuales se debe de planificar el proyecto, la forma angular, la cual indica únicamente el avance del software dentro del proyecto y la forma radial, la cual indica el aumento del costo dado que cada iteración conlleva más tiempo de desarrollo.

Desarrollo por Etapas

Es similar al modelo por etapas, sin embargo, se diferencia en que, al momento de la crítica o bien retro alimentación por parte del usuario final, no se obtendrán completamente las características del software. Estas se irán descubriendo en el proceso del avance del software, mediante la creación de las diferentes versiones del código. En este modelo, se distinguen las siguientes fases:

- Especificación conceptual.
- Análisis de requisitos.
- Diseño inicial.
- Codificación y depuración.

Desarrollo Iterativo y Creciente

Resuelve los problemas encontrados en el modelo en cascada, en cual ofrece entornos de trabajo con técnicas para su correcta utilización. Este tipo de modelo es esencial para el método de programación extrema. Este tipo de programación consiste en la realización de programas de manera incremental, la cual sirve para obtener ventaja de lo que se ha realizado a lo largo del proyecto. En este se entran a varias iteraciones con las cuales se obtendrá el software final y a su vez, se le agregaran nuevas funcionalidades a cada etapa. Se puede dividir en los siguientes procesos:

Etapas de inicialización. La meta de esta etapa es la construcción de un producto en el cual se pueda obtener retroalimentación de parte del usuario final.

Etapas de iteración. Consiste en el análisis, rediseño e implementación del producto de las iteraciones anteriores.

Lista de control del proyecto. Son las tareas que se crean que describen las partes que conforman el proyecto, son implementadas y rediseñadas en cada etapa de iteración del producto.

RAD o Desarrollo de Aplicaciones Rápidas

Como su nombre lo indica permite la construcción rápida de sistemas utilizables. Está compuesto por un grupo reducido de personas incluyendo desarrolladores y testers del sistema. También se debe de hacer énfasis al desarrollo de la aplicación cumpliendo correctamente las funcionalidades principales, dejando a un lado a las implementaciones secundarias. Este modelo toma principalmente en cuenta las características de usabilidad, utilidad y rapidez de la ejecución de la aplicación.

Desarrollo Concurrente

Se conoce como ingeniería concurrente y es utilizado en su mayoría para aplicaciones cliente servidor, en el cual se describen los múltiples procesos que ocurren simultáneamente en la aplicación. Una de las características de este proceso es que está orientado a las necesidades del usuario, las decisiones de la gestión y los resultados de las revisiones. Las ventajas que se pueden mencionar es que está orientado a grupos de trabajo independientes, proporcionando una visión exacta de lo que se lleva desarrollado del proyecto. Las desventajas se tienen que se necesitan de grupos de trabajo y de las condiciones necesarias para su implementación.

Proceso unificado

Este proceso se distingue por la utilización de casos de uso, el cual está centrado en la arquitectura y la utilización de iteraciones incrementales. Este es un marco de trabajo extensible, el cual puede ser implementado hacia otros proyectos de distintas organizaciones. Este proceso es utilizado para evitar problemas legales con el método Proceso Unificado Relacional (RUP) dado que este otro método es una marca registrada de la International Business Machines Corp. (IBM). En cada iteración, se busca el avance e iteración en determinadas áreas, con la cual se obtendrán resultados en los cuales se puede constatar el tiempo que se ha dedicado a las distintas áreas en el desarrollo de software. Sus principales características son: es iterativo e incremental, dirigido por casos de uso, centrado en la arquitectura y enfocado en los riesgos.

RUP o proceso unificado relacional

Es propiedad de IBM desarrollado por la empresa Rational Software, es conocido como RUP y llamado Proceso Unificado Rational, por el nombre de la empresa. Es uno de los modelos más utilizados en el análisis, diseño e implementación de la documentación de sistemas orientados a objetos. Los principios básicos de este método son: adaptar el proceso, equilibrar

prioridades, demostrar valor iterativamente, colaboración entre equipos, elevar el nivel de abstracción y enfocarse a la calidad.

Metodologías de desarrollo de Software

Las metodologías de desarrollo de software se utilizan en el ámbito de la programación, entre otros, con el objeto de trabajar en equipo de manera organizada. Estas metodologías han ido evolucionando a lo largo del tiempo, pasando de ser un mero trámite de organización a ser una base importantísima a la hora de desarrollar software de una manera productiva y eficaz.

En las últimas décadas, las metodologías ágiles de desarrollo de software se han impuesto sobre las demás, tal como indica el último estudio de Project Manager Institute (PMI)., que señala que el 71% de las empresas de ingeniería de software utiliza estas metodologías. Por lo tanto, si quieres dedicar tu carrera profesional al desarrollo de software, deberás aprender cómo funcionan las metodologías ágiles.

Algunas empresas siguen usando metodologías tradicionales de desarrollo de software, que pueden seguir siendo válidas en función del proyecto y de la empresa. Sea como sea, emplear metodologías de desarrollo de software es imprescindible en el ámbito de la programación y similares para generar software de alta calidad.

¿Qué es una metodología de desarrollo de software?

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos organizativos que se aplican para diseñar soluciones de software informático. El objetivo de las distintas metodologías es el de intentar organizar los equipos de trabajo para que estos desarrollen las funciones de un programa de la mejor manera posible.

Cuando se trata de desarrollar productos o soluciones para un cliente o mercado concreto, es necesario tener en cuenta factores como los costes, la planificación, la dificultad, el equipo de

trabajo disponible, los lenguajes utilizados, etc. Todos ellos se engloban en una metodología de desarrollo que permiten organizar el trabajo de la forma más ordenada posible.

El desarrollo de software puede ser un sector especialmente complejo, sobre todo cuando se trata de grandes aplicativos y equipos de trabajo. Ponerse a desarrollar un producto sin una metodología clara desembocará en un proceso aún más complejo, que conducirá a problemas, retrasos, errores y, en definitiva, un mal resultado final.

El trabajo con una metodología de desarrollo de software permite reducir el nivel de dificultad, organizar las tareas, agilizar el proceso y mejorar el resultado final de las aplicaciones a desarrollar.

Maida y Pacienza (2015), exploran a profundidad el concepto de metodología de desarrollo de software, explicando que se trata de una agrupación de técnicas y métodos interrelacionados con el fin de abordar las diferentes fases de un proyecto de desarrollo. En otras palabras, definen que “una metodología de desarrollo de software es un marco de trabajo que se usa para estructurar, planificar y controlar el proceso de sistemas de información”.

Por esta razón, es imperativo realizar un análisis objetivo y completo sobre la metodología que mejor se adapte al proyecto propuesto. Con esto, se pretende que el marco de referencia proporcione una guía simple pero efectiva en aspectos como las fases de desarrollo, actividades a realizar y que documentación se debe generar, con el objetivo de generar un producto de calidad en tiempos factibles.

Maida y Pacienza (2015), señalan que, si un proyecto de software no sigue las directrices de una metodología de desarrollo, o si se escogiese una metodología no optima, existe una baja probabilidad de éxito en el proyecto.

Metodologías Ágiles Contra Metodologías Tradicionales

Uno de los primeros aspectos a considerar en el análisis de las metodologías de desarrollo, es la determinación de la modalidad de la metodología con la que se va a trabajar. Es decir, escoger si la presente investigación se basa en las metodologías ágiles o las tradicionales. Esta sección muestra la diferencia entre las dos para continuar con un estudio más a fondo de lo que conlleva cada metodología.

Metodologías Tradicionales

Molina et al. (2018), en su comparativa sobre metodologías ágiles y tradicionales, los marcos de referencia tradicionales nacieron en la década de los 60 con el propósito de poner un fin a la desorganización causada por el desarrollo de software en grandes cantidades. Proponía una forma de estructurar y definir la ingeniería de software.

Maida y Pacienza (2015), mencionan que, las metodologías de este conjunto tienden a ser más rigurosas, poco flexibles y pesadas, debido a que “centran su atención en llevar una documentación exhaustiva de todo el proyecto, la planificación y control de este, en especificaciones precisas de requisitos y modelado y en cumplir con un plan de trabajo”. Los autores también indican que la fase inicial del proyecto es una de las más importantes, ya que se definen los roles, actividades, artefactos, herramientas y notaciones”.

Metodologías Ágiles

En los años 90, como consecuencia de la dificultad de la aplicación de las metodologías tradicionales, aparecen los marcos de referencia ágiles (o ligeros); García (2015), señala que “esta metodología trata de mejorar la estimación en costes, duración y alcance, de forma que pueda ser adaptable a los cambios y se vuelva una metodología más eficiente, ya que se produce menos documentación”.

Molina et al. (2018) y García (2015), manifiestan que las metodologías ágiles se adaptan mejor a los cambios gracias a su nuevo enfoque. Ya no se trata de una visión orientada a los procesos, sino a las personas, haciendo que el marco de referencia tenga una comunicación constante con el cliente.

Comparación de las Metodologías

En la determinación de la modalidad de metodología de software que se usó para el desarrollo del proyecto, se tomaron como base las comparaciones de García (2015), Maida y Pacienza (2015) y Molina et al (2018). Se consideraron los aspectos más importantes que cada uno de los autores resalta y se utilizaron aquellos que fueran pertinentes para el presente trabajo.

Dentro de estos criterios se encuentran, en orden alfabético, la importancia de la arquitectura de software, la cantidad de artefactos, la adaptabilidad a los cambios, el modelo de ciclo de vida en el que se basa, la modalidad de los contratos, el tipo de cultura organizativa (jerarquía), el enfoque de desarrollo de software, la cantidad de documentación, el número de entregas, el estilo de desarrollo, la orientación de la gestión del proyecto, las metas de los proyectos siguiendo cierta modalidad, el modelo que se emplea, la forma de organización del equipo, la relación con el cliente, las características de los requerimientos, la cantidad de roles en el equipo de desarrollo, y tamaño del equipo de desarrollo.

Tabla 1
Comparación Metodológica

Aspecto	Metodologías Tradicionales	Metodologías Agiles
Arquitectura de Software	La arquitectura es esencial.	No tan enfocado a la arquitectura.
Artefactos	Mayor cantidad de artefactos.	Menor cantidad de artefactos.
Cambios	Resistencia a los cambios.	Facilidad para realizar cambios.
Ciclo de Vida	Secuencial (cascada, espiral).	Iterativo o evolutivo.
Contrato	Contrato prefijado.	Contrato flexible.
Cultura Organizativa	Jerarquía fuertemente establecida.	Menos jerarquía.
Desarrollo de software	Solución universal y segura.	Solución según las necesidades del proyecto.
Documentación	Extensa, detallada y con conocimiento específico.	Reducida, ligera y con conocimiento tácito.
Entregas	Una sola entrega (al finalizar).	Varias entregas constantes de los avances en el software.
Estilo de Desarrollo	Predictivo(anticipado).	Adaptativo.
Gestión	Orientado a procesos.	Orientado al cliente.
Metas	Optimización.	Adaptabilidad.
Modelo	Pesado, rígido y sobredimensionado.	Flexible: sé hace lo que se necesita.
Organización del Equipo	Equipos preestablecidos.	Equipos autoorganizados.
Organización del Proyecto	Un solo proyecto completo.	El proyecto está dividido en varios subproyectos.
Relación con el Cliente	Poca relación con el cliente.	Mucha relación con el cliente (es parte del equipo de desarrollo).
Requerimientos	Conocidos, estables y predefinidos.	Desconocidos, cambiantes y definidos durante el proyecto.
Roles	Muchos roles.	Pocos roles.
Tamaño del Equipo	Grupos grandes.	Grupos pequeños y medianos (menos de 10 personas).

Nota. se presenta el cuadro comparativo entre las metodologías tradicionales y las metodologías agiles, con la respectiva descripción de lo que implica cada aspecto descrito anteriormente, construido a partir de la información recopilada de los autores mencionados anteriormente. Esta

comparación sirve como sustento para la selección fundamentada de la modalidad de metodología que se utilizó en el proceso de construcción del software.

Paradigmas en el Desarrollo de Software

De acuerdo con los lenguajes de programación que se usen, se deben adoptar determinados paradigmas para darle continuidad a l proceso de forma fluida y sin contratiempos. Sin embargo, existen muchos paradigmas de programación y aquí te contaremos 5 de los más importantes.

Paradigma imperativo, también llamado paradigma por procedimientos

Es probablemente, uno de los más conocidos en el mundo de la programación. Como su nombre lo indica, este es un método que permite desarrollar programas a través de procedimientos. Mediante una serie de instrucciones, se explica paso por paso cómo funciona el código para que el proceso sea lo más claro posible.

Paradigma funcional

Una de las características del paradigma funcional es que este, como su nombre lo indica, trabaja a través de determinadas funciones matemáticas. Este es un tipo de paradigma que se usa, principalmente, en el ámbito académico más que en el comercial. A diferencia del paradigma imperativo, aquí importa más el “que” y tanto el “como” se desarrolla un proyecto.

Paradigma declarativo, también conocido como paradigma de programación lógica

El paradigma declarativo es aquel que se preocupa por el resultado final desde el inicio. Determinar de forma automática la ruta a seguir para conseguir una solución puede resultar muy eficaz a la hora de programar, solo se necesita tener claridad en torno al proceso que se va a llevar adelante.

Paradigma reactivo

El paradigma reactivo está enfocado en analizar el flujo de datos, ya sean finitos o infinitos, con el fin de responder a las necesidades que se presenten durante el desarrollo de los proyectos en términos de escalado, y para procurar una reacción inmediata al cambio de valores que se producen por los flujos de datos.

Paradigma orientado a objetos

Este tipo de paradigma de programación ofrece una guía que permite identificar como trabajar con él a través de objetos y planos de código. Este tipo de paradigma se constituye por piezas simples u objetos que al relacionarse entre sí forman diferentes componentes del sistema que estemos trabajando.

Boceto del Software

Saber cómo hacer un boceto de una página web es vital para que el cliente sepa cuál será el resultado final, facilitar las idas y vueltas y prevenir inconvenientes. Si estamos comenzando en el diseño web, plasmar las ideas con anterioridad nos ahorrara horas de trabajo.

El trabajo previo de planificación también es importante ya que te permite detectar posibles problemas y evitar trabajos innecesarios. Además, crear un boceto de una página web te permite definir la jerarquía de la información, lo que favorece su comprensión y usabilidad web.

Como el proceso de diseño de una página web es largo y requiere de investigación exhaustiva, herramientas como los wireframe, que te permiten hacer el prototipado de un sitio web, se volvieron imprescindibles. Básicamente, es un boceto en escala de grises que representa visualmente, de forma muy sencilla y esquemática, cual es la estructura de una página web o de una aplicación móvil.

El boceto de una página web se usa en las primeras etapas del proceso de desarrollo con el objetivo de establecer el esquema antes de agregar el contenido y el diseño visual. Se puede crear en papel, usar lista de etiquetas HTML y propiedades CSS o con aplicaciones para hacer bocetos de páginas web.

Investiga a tu usuario y el diseño de (User Experience) UX

Antes de comenzar a elaborar el boceto de una página web, es necesario investigar a tu público objetivo. Esto puede ayudarte a determinar que funciones deben destacarse más en tu sitio web para que los visitantes puedan encontrar lo que necesitan.

Los User personas pueden ser una herramienta de diseño útil para esta parte del proceso de cómo hacer un boceto de una página web. Por lo tanto, define quienes serán tus grupos de usuarios potenciales, así tendrás una referencia a la que puedas volver durante todo el proceso de diseño del wireframe. Es más, las User personas también pueden ayudarte a crear una estrategia de marketing más adelante.

Define el diseño de tu página web

Lo esencial para aprender cómo hacer un boceto es entender que la página web que estas creando debe ser fácil de usar.

Por eso, antes de diseñar la interfaz (UI) debes planificar la experiencia de usuario (UX) con el fin de aumentar la lealtad y satisfacción del cliente. A estos dos aspectos súmale la arquitectura de la información (AI) de la página web y obtendrás el proceso de diseño del wireframe o boceto completo.

Diseño de información

Para una comunicación clara la información debe estar presentada de manera efectiva, es decir, pensar en la estructura, el contenido y el estilo del diseño en función de los objetivos de la empresa.

La arquitectura de la información (AI) juega un papel fundamental en el boceto de la estructura de una página web, ya que ubica y prioriza la información con el fin de lograr la comprensión por parte de los usuarios.

Crear el boceto de una página web es la mejor forma de obtener comentarios de los usuarios en las primeras etapas de diseño para así evitar rediseños costosos una vez que esté funcionando.

Inclusión de Interfaces Gráficas en el Software

La interfaz gráfica de usuario es un software base que permite la interacción del usuario por medio de elementos gráficos como botones, ventanas, iconos o enlaces, entre otros. La inclusión de la primera interfaz de usuario grafica en una maquina se atribuye a Xerox, con su computadora Xerox Alto a principios de los 70, posteriormente se masifico el uso e inclusión de interfaces graficas en los sistemas operativos de los computadores de uso doméstico y corporativo por parte de Microsoft y Apple, con el auge de la incorporación de gráficos en la interacción de los usuarios con las maquinas, los desarrolladores de software y grandes corporaciones y marcas que se dedicaban a crear soluciones de software también apostaron por implementar esta tecnología, debido a las grandes ventajas que suponga acercar a los usuarios de una manera sencilla al complejo lenguaje de las maquinas.

Esto, significo un vertiginoso avance en el consumo y desarrollo de la tecnología, que aun vemos hoy y que seguirá creciendo en los próximos años, gracias a que permitió disminuir significativamente las brechas que separaban a la tecnología del grueso de los consumidores y

minimiza las curvas de aprendizaje necesarias para el uso de los computadores de la época, que funcionaban a base de comandos que se ingresaban por medio de líneas de texto.

Fue tan importante la inclusión de interfaces graficas de usuarios, que, hoy en día, es una cualidad de facto en la producción de software a excepción de algunos desarrollos muy específicos y que definen índices diminutos comparados con el grueso del software que se desarrolla en la actualidad.

Ben Shneiderman, catedrático en informática de la universidad de Maryland en Estados Unidos define la interfaz como “la membrana de comunicación entre los usuarios y el sistema por la que los diseñadores y desarrolladores de herramientas informáticas pueden hacer que la tecnología sea inteligible y sensible a las necesidades de los usuarios” (1998, p.4).

En este caso en particular, la interfaz gráfica se puede considerar un mediador visual que facilita la comunicación fluida y clara entre un sistema y el usuario para facilitar su comprensión y uso. Esta, por lo general, se manifiesta en un entorno grafico compuesto principalmente de imágenes y objetos para representar la información y las acciones disponibles para ejecutar.

Marco Conceptual

la producción de software es una ciencia compleja y utiliza una serie de criterios, estándares y fundamentos de la ingeniería de software para crear productos industriales y a su vez utiliza bases de la ingeniería como métodos, técnicas y herramientas para desarrollar productos innovadores regidos por metodologías y buenas prácticas, adicionalmente y para el desarrollo del proyecto actual, es necesario comprender ciertos conceptos claves que permiten identificar el público objetivo y sus necesidades.

Ingeniería de requerimientos

Respecto a la definición de un requerimiento es: “es una declaración abstracta de alto nivel de un servicio que debe proveer el sistema o como una restricción de este”. Igualmente es pertinente señalar que la importancia de la ingeniería de requerimientos radica en el aporte que se le da al sistema en cuestión en varios aspectos como la gestión estructurada de necesidades, la mejora en la planificación de cronogramas, disminución de costos, mejoramiento en la calidad del software, así como en la comunicación entre los miembros del equipo, y la prevención de rechazos de parte del cliente (Quintero Vélez, 2016).

Está claro que es de mucha importancia aplicar la ingeniería de requerimientos a la herramienta a desarrollar.

Para poder entender el proyecto que se pretende desarrollar es muy importante saber que es un sistema y sus componentes.

Un sistema es un conjunto de componentes que interaccionan entre sí para lograr un objetivo común. Los recursos acceden al sistema a través de los elementos de entrada para ser modificados en la sección de transformación. Este proceso es controlado por el mecanismo de control con el fin de lograr el objetivo marcado. “Una vez se ha llevado a cabo la transformación, el resultado sale del sistema a través de los elementos de salida”. (Fernández Alarcón, 2018).

Ya sabemos que es un sistema en general, pero debemos conocer a fondo que es un sistema de información: Los autores Laudon y Laudon¹ definen los sistemas de información como un conjunto de componentes interrelacionados que recolectan, procesan, almacenan y distribuyen información para apoyar la toma de decisiones y el control de una organización. Un sistema de información está formado por un conjunto de componentes. Esto es debido a querer englobar los distintos sistemas de información en una única definición. De forma similar, nos podemos encontrar con sistemas de información basados en ordenadores y sistemas de información que utilizan la tecnología del lápiz y el papel.

Los sistemas de información basados en herramientas tanto como de software como de hardware de alta calidad constituyen la parte medular de la mayoría de las corporaciones globales exitosas en la actualidad. Para que una empresa u organización tenga éxito a nivel global, debe ser capaz de proporcionar la data correcta a las personas responsables de la misma en el tiempo correcto, es decir, en momentos oportunos así no se encuentre cerca el destinatario. “La capacidad de un sistema para organizar información de tal forma que esta sea eficiente y eficaz para la toma de decisiones inteligentes constituyen el valor real de los sistemas de información basados en computadora” (M. Stair & Reynolds, 2017).

Todo sistema de información se compone de una serie de recursos interconectados y en interacción, dispuestos del modo más conveniente en base al propósito informativo trazado, como puede ser recabar información personal, procesar estadísticas, organizar archivos, etc. Estos recursos pueden ser:

- Recursos humanos. Personal de variable índole y destrezas.
- Datos. Cualquier tipo de información masiva que precisa de organizarse.
- Actividades. Procedimientos, pasos a seguir, estaciones de trabajo, etc.

- Recursos informáticos. Aquellos determinados por la tecnología.

Los sistemas de información prestan apoyo a los sistemas de actividad humana. Un sistema de actividad humana es un sistema social al que a veces nos referimos como un sistema “soft” o sistema blando. Es una colección lógica de actividades realizadas por algún grupo de personas. Un sistema de actividad humana tendrá uno o varios objetivos claros que cumplir. Otro nombre que se utiliza en la actualidad como un sinónimo para un sistema de actividad humana es un proceso de organización.

Moncada (2014) señala que Las *características* principales de un sistema de información son:

- Disponibilidad de la información de una manera rápida y oportuna.
- Brindar resultados en tiempos de respuesta y seguridad de la información.
- Genera indicadores que permiten analizar la información para realizar una toma de decisiones basada en datos.
- Evita la pérdida de información.
- Mejora el tiempo en la organización de datos evitando la realización de actividades de manera manual.
- Optimización en la búsqueda de la información filtrando la misma según las especificaciones del usuario.
- Permite optimizar los procesos dentro de la compañía.
- Permite la comunicación e interconexión con diferentes aplicativos instalados en la empresa.

Considerando las características de un sistema de información resaltando la importancia en agilizar los procesos de una organización, respondiendo a las necesidades de los usuarios finales y garantizando así el correcto funcionamiento de la aplicación.

Un sistema de información deberá cumplir con cuatro etapas principales por las cuales se producirá la información requerida por cualquier compañía o entidad organizada, esta se administra y ayuda a la toma de decisiones basadas en datos estadísticos y no en subjetividades.

La entrada de datos es el proceso de parametrización y digitación de los datos recopilados los cuales serán procesados dentro del sistema de información.

En el procesamiento de la información los datos ingresados por el usuario se transforman en la información que solicita el usuario; ejecutándose algoritmos para visualizar ese conjunto de datos como por ejemplo realizando cálculos matemáticos para reflejar la situación financiera de la compañía.

El almacenamiento de la información consiste en el registro de la información procesada por el sistema de información en la base de datos administrada por un sistema de gestión de base de datos el cual puede ser licenciado o en algunos casos gratuito.

La salida de información son los resultados de las operaciones realizadas dentro del aplicativo(informes), los cuales son utilizados para analizar la situación de la empresa cuando se requiera; con la misma y acompañada de datos estadísticos que se centraran y basaran en la toma de decisiones soportadas en hechos en tiempo real.

Siempre se deba tener un proceso de retroalimentación para verificar y revisar quien en el sistema de información no se presenten errores de codificación que afecten la información almacenada, esto con el fin de ajustar y validar la confiabilidad del software.

Javier Tuya et al. (2007) Un sistema de información deba cumplir con tres objetivos):

- Brindar información de apoyo para la toma de decisiones.
- Automatización de los procesos.
- Competitividad del desarrollo de este frente al mercado.

Teniendo en cuenta que un sistema de información en su etapa de producción debiera contener el desarrollo de estos tres objetivos para darle un valor agregado a la entidad garantizando la funcionalidad de esta y ayudándola a administrar y fortalecer su regla de negocio para que esta sea competitiva frente a otras compañías, aunque incluso estas tengan implementado una automatización de sus procesos.

Un sistema de información está compuesto por elementos interconectados entre si haciendo del mismo un proceso el cual funcionara de manera correcta y eficaz frente a las necesidades del cliente; dichos elementos son:

- Personas.
- Hardware.
- Software.
- Bases de datos.
- Elementos de comunicación.

El hardware

Es la parte física de un ordenador o sistema informático. Está formado por los componentes eléctricos, electrónicos, electromecánicos y mecánicos, tales como circuitos de cables y luz, placas, memorias, discos duros, dispositivos periféricos y cualquier otro material en estado físico que sea necesario para hacer que el equipo funcione.

El software

Definido como el soporte lógico intangible de las estaciones de trabajo el cual es un conjunto de componentes lógicos necesarios que cuando se ejecutan realizan una tarea específica en relación con los elementos físicos (Hardware) donde interactúan y ejecutan las instrucciones para obtener el resultado esperado (Niño, 2011).

Las bases de datos

Una base de datos es una herramienta para recopilar información y organizar información. Las bases de datos pueden almacenar información sobre personas, productos, pedidos u otras cosas. Muchas bases de datos comienzan como una lista en una hoja de cálculo o en un programa de procesamiento de texto. A medida que la lista aumenta su tamaño, empiezan a aparecer redundancias e inconsistencias en los datos. Cada vez es más difícil comprender los datos en forma de lista y los métodos de búsqueda o extracción de subconjuntos de datos para revisión son limitados. Una vez que estos problemas comienzan a aparecer, una buena idea es transferir los datos a una base de datos creada con un sistema de administración de bases de datos (DBMS).

Anteriormente una base de datos definida como un banco de datos o un Data Banks, eran conjuntos de ficheros organizados los cuales se ejecutaban por lotes de datos haciendo muy compleja su estructura y codificación; en los años setenta se le bautizo con el nombre de base de datos y allí nacieron los sistemas de gestión de bases de datos SGBD (Por sus siglas en ingles).

Un Sistema de Gestión de Bases de Datos (SGBD)

Un sistema gestor de bases de datos (SGBD) o Database Management System (DBMS) es un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos.

Los gestores de datos o gestores de bases de datos permiten administrar todo acceso a la base de datos, pues tienen el objetivo de servir de interfaz entre esta, el usuario y las aplicaciones.

Entre sus funciones se encuentran la de permitir a los usuarios de negocio almacenar la información, modificar datos y acceder a los activos de conocimiento de la organización. Asimismo, el gestor de base de datos también se ocupa de realizar consultas y hacer análisis para generar informes.

Además, los sistemas de gestión de base de datos pueden entenderse como una colección de datos interrelacionados, estructurados y organizados en el ecosistema formado por dicho conjunto de programas que acceden a ellos y facilitan su gestión.

Un SGBD como colección de datos provee la facilidad de los usuarios a la hora de realizar operaciones sobre el sistema, ya sea para la manipulación de datos lenguaje SQL o para la administración de la estructura propia de la base de datos; los mismos se clasifican de acuerdo con sus estructuras o tipos de datos lo cual está definido por el orden jerárquico de la información almacenada en este sistema organizacional en tablas relacionales que están interconectadas entre sí para dar una mayor coherencia evitando subregistros de la misma.

Prieto Espinoza et al, (2005), dicen que el sistema de gestión de bases de datos está definido por un lenguaje SQL(Structured Query Lenguaje), el cual es utilizado para desarrollar aplicaciones en las estructuras de la base de datos por lo que se administra la información con este lenguaje que a su vez permite definir los tipos de datos, consultarlos y modificarlos cumpliendo así una gestión de información en grandes cantidades garantizando su coherencia y fiabilidad para ser usada como activo intangible importante en una organización.

Características de un Sistema de Gestión de Bases de Datos (Universidad de Murcia,2005):

Evitar la redundancia de datos

El sistema de gestión de bases de datos deberá evitar al máximo la redundancia en los datos que se almacenen en el mismo ya que no se debe guardar copia sobre copia de los mismos datos ya que esto ocasionará desperdicio en el almacenamiento de la base de datos.

Consistencia de datos

Cuando se realicen manipulaciones de los datos almacenados en el SGBD se deberá garantizar la integridad de estos ya que al ser una base de datos relacional se tendrá en cuenta las conexiones que tienen las tablas entre sí a la hora de eliminar o actualizar alguna información requerida por el usuario.

Almacenamiento

El sistema de gestión de base de datos garantizara el almacenamiento de los datos ingresados por el usuario lo cual tendrá que ser eficiente en el manejo de estos para que se guarden en sus tablas correspondientes y/o si se presenta alguna eventualidad en los mismos el sistema deberá generar un mensaje de error informándole al usuario de la novedad presentada.

Accesibilidad

El SGBD debe garantizar el acceso de una multiplicidad de usuarios los cuales realizaran diferentes transacciones en tiempo real sobre la base de datos; se deberá tener en cuenta la seguridad e integridad de la información ya que al ser manipulada por varios usuarios al tiempo esta no debe de variar ni mucho menos debe haber una pérdida de esta.

Centralización

Los datos almacenados en el sistema de gestión de base de datos actuarán de una forma centralizada ya que estarán disponibles en un mismo fichero de almacenamiento establecido por el sistema garantizando la disponibilidad de estos. Por lo general las organizaciones cuentan dentro de su planta de empleados con un administrador de base de datos encargado de custodiar el acceso a las mismas.

Integridad de los datos

La información que es almacena en una base de datos debe pertenecer en todo momento con la realidad que representa (Carrillo Ruiz, et al. 2005), por consiguiente el sistema de gestión de base de datos deberá implementar mecanismos de autenticidad de usuarios para evitar la alteración y/o pérdida de la misma dando como resultado recuperaciones por posibles fallas del software y el no acceso del personal el cual previamente no se autorizó para manejar la información almacenada en la base de datos.

Seguridad

El sistema de gestión de base de datos dentro de su funcionamiento primitivo tendrá la opción de definir roles y perfiles de usuario para identificar y controlar el acceso de estos.

Ventajas de un Sistema de Gestión de Base de Datos

Organización

- Por su estructura y funcionamiento se tiene una organización de la información y de fácil acceso para los usuarios en tiempo real.
- Centralización de la información garantizando la integridad brindando una disponibilidad y continuidad al sistema por novedades que se puedan presentar durante la ejecución del SGBD.

- Evita la redundancia de los datos aprovechando al máximo el almacenamiento de la información.
- Optimización de las operaciones relacionadas con el rendimiento del sistema para que el acceso a la información sea ágil y rápida.
- Permite la integridad con nuevas aplicaciones ya que el sistema constantemente se encuentra actualizado permitiendo así la comunicación con otros sistemas facilitando su unión con la misma.

Desventajas de un Sistema de Gestión de Base de Datos

Los sistemas de gestión de bases de datos llegan a ser altamente complejos por su estructura y funcionamiento. En ocasiones no es fácil de entender dichas cualidades o características de estos sistemas.

Costos operacionales

Los SGBD, como administradores de las bases de datos de las organizaciones son los responsables por la estructura e integridad de estas, pero esta tiene un costo alto de licenciamiento ya que se debe de adquirir por el peso de la propia base de datos ya que cuando es un sistema con licencia gratuita por lo general tendrá una restricción en el almacenamiento de la información.

Vulnerabilidad

Ya que toda la información está centralizada en el sistema de gestión de base de datos ponen en un alto riesgo de vulnerabilidad la compañía ya que si se presenta alguna anomalía se podría perder información valiosa.

En ocasiones se puede tornar lento el acceso a la información ya que por su gran volumen es muy probable que este fenómeno se llegue a presentar.

Teniendo en cuenta lo propuesto y haciendo referencia al desarrollo del proyecto de grado es necesario definir que es una aplicación web y cuáles serán las ventajas y desventajas ya que muchas organizaciones prefieren este tipo de aplicaciones por su alto nivel tecnológico y de impacto en la comercialización de un software.

Web

Web es una palabra inglesa que significa telaraña. Se designa como 'la web' al sistema de gestión de información más popular para transmisión de datos a través de internet.

La web es el diminutivo de world wide web o www cuyas tecnologías para su funcionamiento (HTML, URL, HTTP) fueron desarrolladas en el año 1990 por Tim Berners Lee.

Su nombre viene de las siglas **WWW** (Word Wide Web), y no exageramos si decimos que ha sido el mayor avance tecnológico y social del siglo XX y principios del XXI. Su importancia radica en que permite comunicarse virtualmente a cualquier persona del mundo con otra, o con contenidos que le interesen, y todo ello sin implicar un contacto físico directo. Así, no solo se pueden compartir experiencias, conocimientos, bienes económicos...sino que además se puede hacer con millones de personas como receptores potenciales. La web tal como la conocemos se basa en dos grandes principios: es universal (cualquiera puede acceder mientras tenga conexión a internet) y es de acceso gratuito (aunque en la red haya servicios de pago). Solo basta pararse un momento y meditar sobre las implicaciones que esto tiene en la vida de cada uno para darse cuenta de su importancia.

En la actualidad el desarrollo web es fundamental dado que con esta herramienta tecnológica es posible comunicarnos entre regiones y acceder a la información de casi cualquier organización y empresa, así como solicitar un servicio. El desarrollo web en las últimas décadas ha ido transformando e impulsado la evolución de internet, cada día son más grandes los avances que es este se trata, involucrando directamente el ingenio propio del desarrollador en el tema.

Para que los sitios web sean dinámicos y eficientes se deben tener en cuenta aspectos muy específicos y relevantes para el buen uso de esta tecnología, tanto la parte técnica como en el uso de recursos externos, no es igual visualizar una página de internet en la ciudad donde los recursos de transferencia de datos son altos a la visualización de la misma en ambientes rurales donde los recursos son limitados o nulos, es este uno de los grandes riesgos que se enfrentan, que pueden llevar a perder interés en la misma al no ser accesibles, por lo que la planeación es vital para el buen desempeño de la misma en ambientes de diferentes características siendo tan solo este uno de los problemas que pueden afectar la creación de un sitio web.

Marco Contextual

Inicialmente para el año 2006 en la ciudad de Bogotá, Karen Tatiana Chávez Merchán desarrollo un “Plan estratégico para bibliotecas de centros penitenciarios de Bogotá: Un acercamiento desde la biblioteca pública”, trabajo dirigido por el profesor Roberto Téllez Tolosa, donde se logró alcanzar un acercamiento mediante el cual se lograron estrategias que solidificaron las bases para proponer un plan piloto de implementación en bibliotecas en centros penitenciarios; por medio de este trabajo se dio el primer paso en esta línea de investigación lo que hace ver a la universidad de la Salle como la institución pionera en este tipo de investigaciones con enfoque social.

Posteriormente en el año 2009 las investigadoras de la Pontificia Universidad Javeriana Gladys Adriana Bello García y Luz Adriana Martínez Santos realizaron otro trabajo importante en esta línea de investigación donde se enfocaron en la población del centro penitenciario de Acacias Meta, y luego desarrollaron un diagnóstico para identificar debilidades con la línea de investigación y trazaron 9 objetivos nombrados a continuación:

- Manejo de recursos.
- Accesibilidad a la biblioteca.
- Planta física y equipos necesarios.
- Material tecnológico y sistemas.
- Bibliotecólogos y referencistas.
- Manejo de presupuesto.
- Material bibliográfico.
- Servicios bibliotecarios.
- Programas de promoción de lectura.

Estos objetivos se lograron alcanzar a cabalidad permitiendo cumplirlos sin problemas. Tres Años después en el año 2012, la Universidad de la Salle bajo la dirección del profesor Luis Ernesto Pardo Rodríguez desarrollo un “Proyecto para la consolidación de una cultura de lectura en el centro penitenciario de Chiquinquirá, Boyacá” , el cual fue ejecutado por las investigadoras Emilse Paola Martínez y Derlys Enid Rodríguez R; en este trabajo a partir de un diagnóstico se logró proponer estrategias de creación de programas de lectura y otras actividades relacionadas con la misma, lo cual se soportó en un plan operativo para conformar bibliotecas en los centros penitenciarios administrados por el INPEC. Desde el punto de vista legal, el acceso a la educación, conocimiento e información, para las personas privadas de la libertad, se encuentra reglamentado en diversos documentos nacionales e internacionales, entre ellos: Constitución Política de Colombia, Artículo 20, Código Penitenciario y Carcelario (Ley 65 de 1993), Artículo 94, Declaración americana de los derechos y deberes del hombre: Artículo 14, ítem 3, Principios básicos para el tratamiento de los reclusos, La Carta Constitucional para el Lector (1994), Manifiesto de la Federación Internacional de Asociaciones de Bibliotecarios y Bibliotecas (IFLA) con la Organización de las Naciones Unidas para la Cultura, las ciencias y la Educación UNESCO dan las directrices IFLA/UNESCO para el desarrollo del servicio de bibliotecas públicas(1995).

Sin embargo, el INPEC no ha logrado normalizar los procedimientos en torno a la lectura, que regulen los 141 establecimientos penitenciarios y carcelarios de Colombia para la implementación y administración de las bibliotecas, como actividades que aporten el proceso de resocialización y reinserción social.

Esta labor la ha asumido cada establecimiento de forma independiente a falta de profesionales que desarrollen e implementen las herramientas adecuadas para brindar y permitir

a los internos el acceso a la información y al conocimiento a través de la biblioteca y de esta forma mantener un ambiente participativo entre la población interna, así como ser agente revitalizador mientras cumplen su condena.

Por otro lado y haciendo un análisis del funcionamiento de las bibliotecas en los Establecimientos Penitenciarios de algunos de los países Latinoamericanos, se evidencia que similar a Colombia, en la mayoría de éstos y a pesar de la promulgación de Leyes y normativas relacionadas con los servicios de Bibliotecas en los Establecimientos penitenciarios, los esfuerzos han sido escasos por parte del gobierno inclusive de las entidades administradoras de estos Establecimientos, con la justificación de falta de asignación de recursos financieros, así como el incremento de la población interna.

Sin embargo, en algunos países se evidencia la intención y organización de los establecimientos penitenciarios para una mejor prestación de servicios sociales a sus internos como objetivo principal, a pesar de los múltiples factores que impiden el cumplimiento del deber humano y bienestar social que merecen los internos.

Según la UNESCO existen derechos fundamentales para toda persona en los centros penitenciarios, como es el derecho a leer, preguntar y a tener acceso a los recursos educacionales. Es decir, los penales deben proveer programas educativos para alcanzar una educación y capacitación para el desarrollo cultural y reinserción social de sus habitantes. Es más, deben colaborar en la lucha para reducir los altos porcentajes de analfabetismo.

Asimismo, la IFLA da una serie de lineamientos para este tipo de bibliotecas que van desde tener en cuenta: la administración, el acceso (disposición de horarios según necesidades), instalaciones y equipamiento (iluminación, control de clima, etc.), tecnología de la información (tecnología con fines educativos aunque sea de forma vigilada), personal (debe contar con

profesional bibliotecólogo según las demandas de los reclusos), el presupuesto (debe contar con un presupuesto anual), material bibliográfico (debe ser procesado además de ser en diferentes formatos y de temática variada), servicios y programas (según las necesidades de los reclusos), entre otros.

La promoción de lectura es una práctica de intervención social, en la cual se comparte la palabra escrita a través de la literatura. En Colombia, existen diferentes instituciones, organizaciones y entidades tanto privadas como públicas dedicadas a la promoción de lectura tales como Biblored, Red Nacional de Bibliotecas, Fundalectura, Cerlac, Entre libros y Lectores, Gente y Cuentos, bibliotecas municipales e institucionales, entre otras.

En el documento Lineamientos para la promoción de la lectura de la Red de Bibliotecas del Banco de la República en el cual determina que “diseñar estrategias de intervención lectora se debe considerar a todas las esferas de la población, dedicando especial atención a aquellos grupos sociales a los que generalmente está negado el acceso a la lectura y a los libros” (2010, p. 10). A partir de esta intención, la literatura se convierte en una herramienta al servicio de la sociedad, en especial de personas que se encuentran en condiciones de vulnerabilidad como los 23 desplazados, los internos de las cárceles, y personas hospitalizadas. Es allí donde la literatura cobra mayor valor, puesto que alcanza un vínculo humanizante. Al servicio de estas comunidades, la literatura es una apuesta por dar voz al que no la tiene en esta sociedad, ya que permite la creación de espacios de reflexión y libertad, en especial de las personas en condiciones de vulnerabilidad.

Según Didier Álvarez, uno de los principales teóricos de la promoción literaria en Colombia:

Humanizar con la lectura y la escritura no significa la aceptación de una identidad abstracta del escolar, sino la necesidad de afirmar una vida que debe aprenderse a vivir con dignidad, esperanza y trascendencia en medio de la incertidumbre y la contingencia del mundo, entre otras vías, por la de leer y el escribir. (Álvarez, 2012, p. 90).

En la promoción de Lectura, la lectura y la escritura son procesos dinámicos en los que se encuentran un sin fin de posibilidades, y en ese transcurrir hay un reconocimiento del otro, una reafirmación de los seres humanos. Por su parte, Suarez (2010) propone “la lectura y la escritura como práctica sociocultural que permite a los lectores asumir posturas reflexivas, críticas y comprensivas sobre sus realidades individuales y colectivas, así como generar cambios en sus relaciones y contextos personales y sociales” (p.20). La literatura es una herramienta de transformación de la sociedad que le permite al individuo una apertura valorativa y ética hacia los demás, en la búsqueda del reconocimiento del otro y de sus diferencias.

Teniendo en cuenta los lineamientos de cada una de las organizaciones gubernamentales y no gubernamentales en cuanto a brindar espacios y herramientas que faciliten el acceso a la lectura de las PPL en Colombia es que nació la iniciativa de presentar este proyecto encaminado a desarrollar una herramienta que facilite y le de acceso a los privados de la libertad a los recursos literarios en los centros de reclusión del país es por esto que tomamos como población muestra para el desarrollo de este proyecto de grado el CPAMS Girón en el cual existe una biblioteca con abundante material bibliográfico pero con dificultades para brindar un servicio óptimo que garantice el acceso a la lectura y la protección del inventario así como monitorear la actividad de lectura de los penados.

La producción de software en la actualidad es una profesión que está encaminada a facilitar la vida de las personas que utiliza una serie de criterios, estándares y fundamentos de la

ingeniería de software para crear productos innovadores regidos por metodologías y buenas prácticas, adicionalmente y para el desarrollo del proyecto actual, es necesario comprender ciertos conceptos claves que permitirán identificar la población objeto y sus necesidades.

Delimitaciones

La propuesta ha sido diseñada para implementarse en los centros de reclusión que cuentan con área de educativas, equipos de cómputo y bibliotecas las cuales por motivos de seguridad no están totalmente a disponibilidad de los internos ya que hay restricciones en cuanto al acceso al material bibliográfico, es por esto que se tomó como referente la biblioteca del CPAMS Girón Santander la cual cuenta con abundante material bibliográfico y está ubicada en el área de educativas del penal la cual cuenta con salas de cómputo que le daría acceso a las PPL a él aplicativo web y hacer las consultas y préstamos y de esta manera acceder a el material bibliográfico.

Delimitación Tecnológica

Debido a que se trata de una solución de software, es necesario contar con algún dispositivo tecnológico para visualizar e interactuar con el programa y teniendo en cuenta que se busca tener disponibilidad y accesibilidad a el inventario de la biblioteca, este software y sus datos se encuentran alojados en el servidor del área de educativas, sin embargo, la conexión a internet no será obstáculo ya que primordialmente el aplicativo será utilizado en el área de educativas del CPAMS Girón. El internet se necesitará para la conexión por fuera del establecimiento carcelario solo para dar a conocer al público externo las actividades de lectura que ese realizan en el penal.

Delimitación de la Metodología de Desarrollo de Software

En un primer momento, se debe establecer una guía para la construcción de la aplicación propuesta en el presente trabajo de titulación, para lo cual fue preciso determinar la metodología de desarrollo de software, como también tres marcos de referencia tentativos en base a criterios definidos, a fin de poder precisar el mejor acercamiento para el diseño del sitio web en

específico. Al final del apartado se encuentra una comparativa entre las tecnologías junto con la conclusión respectiva.

El proceso de construcción del software comienza con la identificación de la metodología y del ciclo de vida o fases.

Las Metodologías de Desarrollo De Software

Aplicación al Proyecto

Una vez ya sea han definido los aspectos de comparación entre las modalidades de las metodologías de desarrollo de software, y con las diferencias clarificadas en base a cada uno de estos criterios, es preciso adaptar el análisis hacia el proyecto a desarrollar. Para esto se utilizaron los mismos criterios de comparación de la tabla 1, mostrando cual es más pertinente para el desarrollo de la web. Cabe recalcar que se consideraron todos los aspectos igual de importantes y, por tanto, la ponderación de cada uno fue la misma. Así, la modalidad que tenía la mayor cantidad de características relevantes al proyecto fue la que determino que tipo de metodología se usó en el mismo.

Tabla 2

Selección de Metodología Por Cada Aspecto de la Comparación

Aspecto	Metodología Tradicional	Metodología Ágil	Explicación
Arquitectura de Software		x	Al tratarse de una aplicación web, la arquitectura de la aplicación no es el aspecto primordial en la construcción del software.
Artefactos		x	Los desarrolladores de aplicaciones web no cuentan con grandes cantidades de artefactos. Los framework se encargan de reducir ese número.
Cambios	x		Una definición anticipada de los requerimientos teniendo en cuenta el software existente y la necesidad de cambiarlo por una aplicación

			web que le de acceso directo a los lectores a los contenidos.
Ciclo de Vida	x		El proyecto podría adaptarse a un ciclo de vida secuencial, a causa de los pocos cambios y pocas entregas que se conceptualizaron en un primer momento.
Aspecto	Metodología Tradicional	Metodología Ágil	Explicación
Contrato		x	Es necesario un contrato flexible (o no tradicional), debido a que es un proyecto propuesto y no una contratación de un servicio.
Cultura Organizativa		x	Gracias a la baja cantidad de desarrolladores involucrados en el equipo, una organización poco jerárquica es más útil.
Desarrollo de software		x	Al definir los requerimientos con anticipación, se puede generar una solución diseñada a las necesidades del proyecto, para optimizar tiempo y recursos que se gastarían en una solución universal.
Documentación		x	Un proyecto con documentación extendida requiere de más tiempo asignado. Con el tiempo limitado del presente trabajo, es más adecuada una documentación ligera.
Entregas	x		Debido a que el proyecto fue propuesto, y no una necesidad establecida, no habría mucho interés en varias entregas constantes.
Estilo de Desarrollo	x		Los requerimientos, al ser predefinidos, suponen un estilo de desarrollo predictivo.
Gestión	x		Una relación limitada con el cliente fomenta que la gestión del proyecto sea enfocada al proceso a ser desarrollada en lugar de centrada en el cliente.
Metas		x	El paradigma móvil de programación tiene un enfoque hacia la adaptabilidad con varios dispositivos, aunque al proyecto se le aplicaría a futuro ya que los usuarios son PPL.

Modelo		x	Con un desarrollo enfocado a una solución específica al proyecto, es oportuno mantener un modelo flexible.
Organización del Equipo		x	Los equipos autoorganizados son ventajosos para equipos pequeños de trabajo, debido a que no hay muchas personas para una jerarquía establecida.
Organización del Proyecto	x		Se pensaron pocas entregas de software, y debido a que el trabajo no es muy grande, es útil mantener al software organizado en un solo proyecto completo.
Relación con el Cliente	x		Tomando en cuenta que el proyecto fue un trabajo propuesto, y no una necesidad establecida, es posible que el cliente no esté muy sumergido en el proyecto.
Requerimientos	x		Dado a que el cliente no propuso el proyecto, se debieron establecer los requerimientos con anterioridad. No aportaría mucho al proyecto el definirlos durante la construcción de la aplicación.
Roles		x	A causa a que el equipo de desarrollo tuvo pocas personas involucradas, es necesario tener roles a ser repartidos.
Aspecto	Metodología Tradicional	Metodología Ágil	Explicación
Tamaño del Equipo		x	El software por construir no es muy grande y no hay mucha necesidad de grupos grandes de más de 10 personas.
Total	8	11	

Nota. Ahora considere la tabla 2, que contiene la opción escogida de entre las dos modalidades de metodologías de desarrollo de software, por cada uno de los aspectos de comparación, con la respectiva justificación de dicha selección. Al final del cuadro, se encuentra el total de características preferidas por cada tipo de metodología.

Basándose en los resultados de la tabla 2, se puede inferir que la opción para el desarrollo del proyecto es hacer uso de las metodologías ágiles, por su flexibilidad, ligereza y recurso humano reducido. también se puede apreciar que la diferencia entre las dos modalidades no fue muy grande y, por tanto, se podría también hacer uso de una metodología tradicional. Cabe destacar que la principal razón por la cual se selecciona más veces una metodología ágil que una tradicional, es por limitantes en los recursos (como el tiempo o el capital humano).

De modo que, si se contara con los recursos suficientes, la diferencia entre los dos tipos de metodologías sería aún menor.

De la misma forma, es importante mencionar que, aplicar una metodología ágil no trae inconvenientes en cuanto a los ítems correspondientes a la modalidad tradicional. El hecho de que el trabajo fue una solución propuesta, y no una necesidad identificada por la organización, sugiere que se defina con anterioridad los requisitos de la aplicación para que el cliente esté al tanto de lo que se ira a desarrollar, provocando que el proyecto tienda a ser reacio al cambio, existan pocas entregas o se tenga un estilo predictivo de desarrollo, características que se complementarían con las metodologías tradicionales. Sin embargo, no significa que el cliente vaya a tener una relación distante con la solución propuesta. Por el contrario, se puede motivar a sostener una comunicación activa con la organización, como lo recomienda la modalidad de metodologías ágiles, haciendo que:

- Los requerimientos se puedan definir durante el proyecto,
- Se dé la posibilidad a cambios en el software,
- Se trabaje con un estilo de desarrollo más adaptivo a dichos cambios,
- Se emplee un ciclo de vida evolutivo para un modelo adaptativo y, en consecuencia,

- Se organice el proyecto en subdivisiones con varias entregas de los avances.

De este modo, pese a que algunos criterios se apegan más a una modalidad tradicional de desarrollo, el proyecto fácilmente se puede adaptar completamente a una metodología ágil sin ninguna desventaja. Así, por lo tanto, se concluye que el marco de referencia que se adapta mejor al presente proyecto es referente a una metodología ágil.

Para el desarrollo de este trabajo se asume un enfoque de desarrollo tecnológico basado en el ciclo de vida de la metodología SCRUM (Sutherland, 2012).

Arquitectura de Software Modelo Vista Controlador

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

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

El Modelo

Que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

La Vista

Interfaz de usuario que compone la información que se envía al cliente y los mecanismos interacción con este.

El Controlador

Que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El modelo es el responsable

De acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.

Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser. “si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.

Lleva un registro de las vistas y controladores del sistema.

Si estamos ante un modelo activo, notificara a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).

El controlador es responsable

De recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).

Contiene reglas de gestión de eventos, del tipo “Si evento Z, entonces Acción W” estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método “Actualizar ()”. Una petición al modelo puede ser “obtener_tiempo_de_entrega (nueva_orden_de_venta)”.

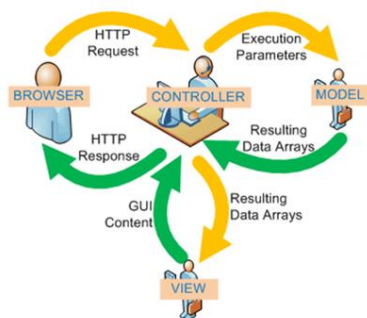
Las vistas son responsables de

Recibir datos del modelo y los muestra al usuario.

Tiene un registro de su controlador asociado (normalmente porque además lo instancia). Puede dar el servicio de “Actualización ()”, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

Figura 1

Flujo Del Modelo Vista Controlador



Nota. La imagen nos muestra el flujo que sigue cada uno de los elementos que integran el modelo vista controlador.

Para complementar el modelo de desarrollo seleccionado, se establece el modelo Vista Controlador (MVC), como modelo de arquitectura de software base, donde se separan los datos de la aplicación o el Software, la Interfaz de Usuario y la lógica de control en tres componentes distintos, esto también permite mayor maniobrabilidad y flexibilidad para su desarrollo, además de tratarse de un modelo muy maduro y que ha demostrado su validez a lo largo de muchos años en todo tipo de aplicaciones, en una gran cantidad de lenguajes y plataformas de desarrollo, incluyendo por supuesto las plataformas web.

Patrones de Diseño de Software

¿Qué es un Patrón de diseño?

Los patrones de diseño son soluciones habituales a problemas que ocurren con frecuencia en el diseño de software. Son como planos prefabricados que se pueden personalizar para resolver un problema de diseño recurrente en tu código.

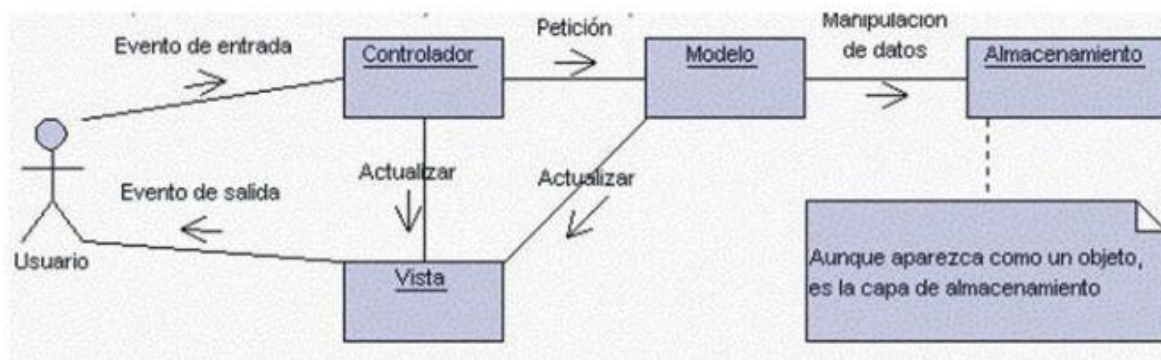
No se puede elegir un patrón y copiarlo en el programa como si se tratara de funciones o bibliotecas ya preparadas. El patrón no es una porción específica de código, sino un concepto general para resolver un problema particular. Puedes seguir los detalles del patrón e implementar una solución que encaje con las realidades de tu propio programa.

A menudo los patrones se confunden con algoritmos porque ambos conceptos describen soluciones típicas a problemas conocidos. Mientras que un algoritmo siempre define un grupo claro de acciones para lograr un objetivo, un patrón es una descripción de más alto nivel de una solución. El código del mismo patrón aplicado a dos programas distintos puede ser diferente.

Una analogía de un algoritmo sería una receta de cocina: ambos cuentan con pasos claros para alcanzar una meta. Por su parte, un patrón es más similar a un plato, ya que puedes observar cómo son su resultado y sus funciones, pero el orden exacto de la implementación depende de ti.

Incorporar patrones de diseño en el proyecto, supone una gran optimización de los recursos, ya que estos patrones de diseño se conocen por ser soluciones generales, reutilizables y aplicables a diversas problemáticas de diseño de software. Generalmente, el uso de patrones de diseño evita volver constantemente a la etapa primaria de desarrollo, ya que se alimenta de errores previos de experiencias de otros desarrolladores, que ensamblaron una serie de plantillas que ayudan a identificar condiciones de error y problemas en el código que pueden no ser evidentes en un modelo dado, son soluciones que funcionan y que han sido usados y probados por una gran cantidad de desarrolladores, haciendo que este tipo de proyectos sean menos propensos a errores, actualmente se conocen hasta 23 patrones de diseño, clasificados en tres categorías: Creacional, Estructural y de Comportamiento.

Figura 2
Patrón Modelo Vista Controlador



Nota. Interrelación entre los elementos del patrón MCV

Revista Digital De Tecnologías De La Información Universidad Tecnológica De La

Habana; Yanette Díaz, Yenisleidy Fernández. (2012). Patrón Modelo-Vista-Controlador.

Herramientas

Framework para Desarrollo de Aplicaciones Web Laravel

Laravel es un marco de aplicación web con una sintaxis expresiva y elegante. Un marco web proporciona una estructura y un punto de partida para crear su aplicación, lo que le permite concentrarse en crear algo increíble mientras nos preocupamos por los detalles.

Laravel se esfuerza por brindar una experiencia de desarrollo increíble al mismo tiempo que brinda características poderosas, como una inyección de dependencia completa, una capa de abstracción de base de datos expresiva, colas y trabajos programados, pruebas unitarias y de integración, y más.

¿Por qué utilizar laravel en el proyecto?

Es estable. Este framework permite agregar información de utilidad mediante su directorio Packalyst. Es eficiente de resolver de manera inmediata aquello que haga falta para la resolución de algún problema.

Es compatible. Laravel se compone de un sistema de enrutamiento eficaz, lo que permite relacionar elementos de una determinada aplicación con las rutas que el uso introduce en el navegador. También, permite el desarrollo de microservicios y aplicaciones.

Es actualizable. Laravel permite hacer las actualizaciones de sus versiones sin ningún problema.

La curva de aprendizaje es fácil por lo que posibilita hacer este proyecto; como se menciona es fácil desde el enrutamiento de las páginas y la instalación de herramientas externas a esta.

Costo. Para el desarrollo de este proyecto es gratuito las librerías y herramientas externas necesarias para que el proyecto se lleve a cabo con éxito y cumpla con los objetivos planteados con anterioridad.

MySQL

MySQL es un sistema de gestión de base de datos relacional desarrollado bajo licencia dual: Licencia publica 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.

Compatible

Para la herramienta esta base de datos es totalmente compatible. En Laravel de hecho viene por defecto por lo que ayudara mucho en la programación de la aplicación. La base de datos de la biblioteca, aunque está diseñada en SQL server es de fácil migración a MySQL.

Actualizable

Esta base de datos lleva varios años en el mercado por lo que tiene un soporte bueno y el factor de tiempo no sería un factor negativo. Además de esto tiene varias interfaces de usuario, lo cual es de gran ayuda.

Costo

MySQL no tiene ningún costo es gratuito solo tiene costo si hay la necesidad de alojar la base de datos en la nube.

Tabla 3
Ventajas y Desventajas de MySQL

Ventajas	Desventajas
MySQL es de uso libre y gratuito.	Al ser software libre, muchas de las soluciones para las deficiencias del software no están documentados ni presentan documentación oficial.
No se necesita disponer de hardware o software de alto rendimiento para la ejecución del programa.	Muchas de sus utilidades tampoco presentan documentación.
Velocidad al realizar las operaciones y buen rendimiento.	Se debe controlar/monitorear el rendimiento de las aplicaciones en busca de fallos.
Facilidad de instalación y configuración.	No es el más intuitivo de los programas que existen actualmente para todos los tipos de desarrollos.
Soporte en casi el 100% de los sistemas operativos actuales	No están eficaz en aplicaciones que requieran de una constante modificación de escritura en BD.

Nota. En esta tabla podemos apreciar la comparación de las ventajas y desventajas de la herramienta MySQL.

XAMPP

XAMPP es una herramienta de desarrollo que te permite probar tu desarrollo web basado en PHP en tu propio ordenador sin necesidad de tener acceso a internet. Si eres un diseñador web o desarrollador web que recién está comenzando, no es necesario saber sobre las configuraciones de servidores (aun), ya que XAMPP te provee de una configuración totalmente funcional desde el momento que lo instalas. Básicamente lo extraes y listo. Es bueno acotar que la seguridad de datos no es su punto fuerte, por lo cual no es suficientemente seguro para ambientes grandes o de producción.

¿Pero, Que Es XAMPP?

XAMPP

Es una distribución de Apache que incluye diferentes softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen:

Linux

Es el sistema operativo donde estará instalado nuestra aplicación. A diferencia de Windows, Linux es una distribución libre que es segura, no requiere pago de licencias y tiene alto rendimiento.

Apache

El servidor web de código abierto es la aplicación usada globalmente para la entrega de contenidos web. Las aplicaciones del servidor son ofrecidas como software libre por la Apache Software Foundation.

MySQL/MariaDB

XAMPP cuenta con uno de los sistemas relacionales de gestión de bases de datos más populares del mundo. En combinación con el servidor web Apache y el lenguaje PHP, MySQL sirve para el almacenamiento de datos para servicios web. En las versiones actuales de XAMPP esta base de datos se ha sustituido por MariaDB.

PHP

Es un lenguaje de programación de código de lado del servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente de plataforma y soporta varios sistemas de bases de datos.

Perl

Este lenguaje de programación se usa en la administración del sistema, en el desarrollo web y en la programación de red. También permite programar aplicaciones web dinámicas.

Visual Studio Code

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.

Visual Studio Code se basa en Electron, un *framework* que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el *framework* Electrón, el software no usa Atom y en su lugar emplea el mismo componente editor (Mónaco) utilizado en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

Características

Visual Studio Code es un editor de código fuente construido sobre el *framework* Electron. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un lenguaje dado, como se muestra en la siguiente tabla. Muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos. json (por ejemplo, preferencias del usuario). La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo que está fuera de él. Esto también se aplica a los comandos que requieren mucho tiempo. Cuando esto sucede, el comando en progreso se cancela.

En el rol de editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una opción entre LF y CRLF) y el lenguaje de programación del documento activo.

Recientemente, se lanzó al público la versión web del editor bajo el dominio `vscode.dev`. A diferencia de la versión de escritorio, esta no hace uso de Electron, no posee compatibilidades con ciertos complementos y no se puede acceder al terminal integrado que posee.

Tecnologías utilizadas en el desarrollo del proyecto

HTML

HTML, siglas de HyperText Markup Language (Lenguaje de marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para completar el texto con objetos tales como imágenes. HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>).

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado Word Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio W3C define el lenguaje HTML como “un lenguaje reconocido universalmente y que permite publicar información de forma global”. Por convención, los archivos de formato HTML usan la extensión `.htm` o `.html`.

CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuarios o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "Style".

PHP

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Es también el módulo Apache más popular entre los computadores que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server Firebird y SQLite.

PHP es una alternativa a las tecnologías de Microsoft ASP Y ASP.NET, a ColdFusion de la compañía Adobe, a JSP/Java de Sun Microsystems, y a CGI/Perl.

Modelo Cliente-Servidor

Figura 3

Modelo Cliente Servidor



Nota. Esta imagen nos ilustra cómo funciona del protocolo de transferencia de Hipertexto (HyperText Transfer Protocolo) el cual articula los intercambios de información entre clientes web y los servidores HTTP.

La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como World Wide Web.

Desde el punto de vista de las comunicaciones, esta soportado sobre los servicios de conexión TCP/IP, y funciona de la misma forma que el resto de los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (Documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL.

El servidor web es el responsable de atender las solicitudes y/o peticiones que realiza el cliente por medio de diferentes protocolos generalmente HTTP; los mismos están formados por paginas HTML y scripts ejecutables al lado del servidor los cuales ayudan a cumplir dicha tarea de peticiones. (María Estela Raffino, 2020).

Java Script

JavaScript (abreviado comúnmente 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.

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.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio de 2015.

JavaScript se diseñó con una sintaxis similar a C++ y Java, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen

semánticas y propósitos diferentes. Su relación es puramente comercial, tras la compra del creador de Java (Sun Microsystems) de Netscape Navigator (creador de LiveScript) y el cambio de nombre del lenguaje de programación.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM). JavaScript es el único lenguaje de programación que entienden de forma nativa los navegadores.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

Base De Datos

SQL

El lenguaje de consulta estructurado (SQL Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del algebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre la misma. Una de sus características es el manejo del algebra y el cálculo relacional permitiendo lanzar consultas con

el fin de recuperar información de interés de una base de datos, de una forma sencilla. Es un lenguaje de cuarta generación (4GL).

Seguridad Informática

La seguridad informática, también conocida como ciberseguridad, es el área relacionada con la informática y la telemática que se enfoca en la protección de la infraestructura computacional y todo lo vinculado con la misma, y especialmente la información contenida en una computadora o circulante a través de las redes de computadoras. Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas, y leyes concebidas para minimizar los posibles riesgos a la infraestructura y/o a la propia información. La ciberseguridad comprende *software* (bases de datos, metadatos, archivos), *hardware*, redes de computadoras, y todo lo que la organización entienda y valore como un riesgo si la información confidencial involucrada pudiera llegar a manos de otras personas, por ejemplo, convirtiéndose así en información privilegiada. (Bishop, 2003).

La definición de seguridad de la información no debe ser confundida con la de «seguridad informática», esta última solamente se encarga de la seguridad en el medio informático, pero, por cierto, la información puede encontrarse en diferentes medios o formas, y no exclusivamente en medios informáticos.

La seguridad informática también se refiere a la práctica de prevenir los ataques maliciosos, a las computadoras y los servidores, a los dispositivos móviles, a los sistemas electrónicos, a las redes y los datos, etc.

La seguridad de la información nos habla sobre métodos y procesos que procuran proteger los archivos de información en sus diferentes formas y estados, mientras que la seguridad informática nos habla sobre métodos y procesos técnicos para la protección de la confidencialidad, disponibilidad e integridad de la información.

La seguridad informática también se refiere a la práctica de prevenir los ataques maliciosos, a las computadoras y los servidores, a los dispositivos móviles, a los sistemas electrónicos, a las redes y los datos, etc.

En resumen, la seguridad en un ambiente de red es la habilidad de identificar y eliminar vulnerabilidades. Una definición general de seguridad debe también poner atención a la necesidad de salvaguardar la ventaja organizacional, incluyendo información y equipos físicos, tales como los mismos computadores. Nadie a cargo de seguridad debe determinar quién y cuándo puede tomar acciones apropiadas sobre un ítem en específico. Cuando se trata de la seguridad de una compañía, lo que es apropiado varía de organización en organización. Independientemente, cualquier compañía con una red debe tener una política de seguridad que se dirija a la conveniencia y la coordinación.

Lenguaje Unificado De Modelado(UML) e Ingeniería De Software

Lenguaje unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un “plano” del sistema (modelado), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un “lenguaje” para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar

soporte a una metodología de desarrollo de software (tal como Proceso Unificado Racional o RUP), pero no especifica en sí mismo que metodología o proceso usar.

UML usa elementos y los asocia de diferentes formas para formar diagramas que representan aspectos estáticos o estructurales de un sistema, y diagramas de comportamiento, que captan los aspectos dinámicos de un sistema.

Los principales diagramas son:

- Diagrama de clases.
- Diagrama de objetos.
- Diagramas de casos de uso.
- Diagrama de estados.
- Diagrama de secuencias.
- Diagrama de actividades.

¿Porque Utilizar UML en el Proyecto?

El diagrama de distribución UML muestra la arquitectura física de un sistema informático. Puede representar a los equipos y a los dispositivos, y también mostrar sus interconexiones y el software que se encontrara en cada máquina. La asociación se refiere a la conexión física entre los nodos, como por ejemplo Ethernet, es por esto por lo que para la implementación eficaz del proyecto propuesto es clave el buen manejo de esta herramienta y conocerlo a fondo.

Diseño y Desarrollo Del Proyecto

Modelo De Desarrollo

Metodología SCRUM

Manifiesto Ágil. El Agile Manifiesto o manifiesto para el desarrollo de software ágil es una proclamación de cuatro reglas vitales y doce principios que sirven de introducción y guía para las personas en el software de gestión ágil.

Existen cuatro reglas vitales dentro de este manifiesto:

Individuos e interacciones por encima de procesos y herramientas. La metodología agile quiere asegurarse de que los miembros del equipo trabajan unos con otros, colaboran y obtienen en común el mejor de los posibles resultados.

Adicionalmente, en los equipos agile es muy importante evaluar las opiniones individuales y la creatividad.

Trabajar con el software por encima de la documentación. Este principio significa que el equipo debería estar invirtiendo tiempo en trabajar en cosas que realmente produzcan valor para el cliente y evitar invertir tiempo y recursos en otras actividades como debatir, escribir o revisar documentación.

Colaboración con el cliente por encima de la negociación contractual. Esto no significa que en el framework de agile no existan los contratos entre ambas partes, sino tratar de que sean contratos flexibles que permitan adaptación al cambio y enfocar el tiempo en lo que realmente importante: crear producto y entregar valor a los usuarios.

Responder al cambio por encima de seguir con un plan. La planificación del proyecto debe ser abierta y flexible, esto supondrá que el proyecto pueda responder mejor a los cambios a lo largo del proyecto, ya sean de tecnología o de requisitos, etc.

¿Qué es Scrum?

Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Al igual que un equipo de rugby (de donde proviene su nombre) cuando entrena para un gran partido, scrum anima a los equipos a aprender a través de las experiencias, a autoorganizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente.

Aunque son los equipos de desarrollo de software los que utilizan con mayor frecuencia este tipo de scrum, sus principios y lecciones se pueden aplicar a todo tipo de trabajo en equipo. Esta es una de las razones por las que es tan popular. Aunque se considera a menudo un marco de gestión de proyectos ágil, scrum incluye un conjunto de reuniones, herramientas y funciones que, de forma coordinada, ayudan a los equipos a estructurar y gestionar su trabajo.

Roles Scrum

En Scrum existen 3 roles principales:

Dueño del producto (Product Owner). El Product Owner o dueño del producto es conocido como “La voz del cliente”, ya que debe entender las necesidades y prioridades de los stakeholders, incluyendo los clientes y los usuarios. Este rol es responsable, entre otras cosas, de garantizar una comunicación clara sobre el producto, de definir criterios de aceptación y asegurar que se cumplan. Dicho de otra forma, el Product Owner es responsable de asegurar que el equipo Scrum ofrezca valor.

ScrumMaster. El Scrum Máster debe tener una actitud de líder servicial (Servant Leader) para el equipo Scrum. Es quien modera y facilita las interacciones del equipo como facilitador y motivador. Este rol es responsable de asegurar un ambiente de trabajo productivo

para el equipo, debe protegerlo de influencias externas, despejar los obstáculos y garantizar que se cumplan los principios, aspectos y procesos de la metodología Scrum.

Equipo de desarrollo (Scrum Team). El Scrum Team también conocido como equipo de desarrollo, ya que es responsable del desarrollo, servicio o de cualquier otro resultado. Consiste en un grupo de personas que trabajan en las historias de usuario en la lista de pendientes del sprint para crear los entregables del proyecto. Nadie, ni siquiera el Scrum Máster, indica al Scrum Team como cumplir los objetivos del sprint, es un equipo auto gestionado y multifuncional que cuenta con todas las habilidades necesarias. El tamaño óptimo de un equipo scrum es de seis a diez miembros.

Elementos Scrum

Los elementos que forman el Scrum son:

- **Product Backlog:** Lista de necesidades del cliente.
- **Spring Backlog:** Lista de tareas que se realizan en un Spring.
- **Incremento:** Parte añadida o desarrollada en un Spring, es una parte terminada y totalmente operativa.

Product Backlog. Aquí se almacenan en forma de lista todos los requisitos y funcionalidades en orden de prioridad. Estos requisitos o funcionalidades serán los que tendrá el producto final o los Irán adquiriendo en las sucesivas iteraciones.

Las tres características principales de la lista de objetivos serán:

- En la lista se indican las posibles iteraciones y los releases (lanzamientos).
- La lista ha de incluir los posibles riesgos e incluir las tareas necesarias para solventarlos.

- Contendrá los objetivos del producto, y el valor que le da el cliente y el coste estimado, priorizando por valor y coste.

En el primer Spring es necesario que se definan cuáles van a ser los objetivos del producto y tener la lista de requisitos ya definida. Después de definidos los requisitos se tienen que acordar cuando un objetivo esta completado. Se entiende que el producto está terminado cuando se asegura que se puede entregar una demostración de los requisitos que ya se han cumplido y se incluye que está realizando todo lo que el cliente desea.

Spring Backloog. Durante la planificación de un Spring se elabora una lista de tareas que desarrollara cada equipo, así como el tiempo para terminarlas. De esta manera el proyecto se descompone en pequeñas tareas y revisar si está avanzado, y en caso contrario determinar porque no se avanza e intentar eliminar el problema.

Generalmente para llevar el control de las tareas se usan pizarras, hojas de cálculo o herramientas colaborativas, a cada objetivo se le asignan tareas y se van moviendo de una columna a otra para cambiar el estado. Se debe incluir la lista de tareas, la persona responsable de cada tarea, el estado en el que se encuentra y el tiempo restante para completarla.

Incremento. Aquí se muestran los requisitos que se han completado en cada iteración y que son operativos. Según esto el cliente puede hacer cambios y replantear el proyecto.

Dinámica

Iteraciones

Son conocidas como Sprints, y como toda metodología ágil es incremental e iterativo. Esto significa que el producto se construye en incrementos funcionales que se entregan en periodos cortos. En general se recomienda que cada Sprint sea entre 1 o 4 semanas, siendo lo más habitual 2 o 4 semanas.

Reunión de planificación de Sprint

En cada Sprint se realiza una reunión en donde se generan los acuerdos y compromisos del equipo de desarrollo y el dueño del producto sobre el alcance del Sprint. Normalmente la reunión se divide en dos partes, el que se va a realizar y el cómo se hará.

En la parte del “que”, básicamente el Dueño del Producto expone todos los product Backlogs que podrían formar parte del Sprint, y el equipo de desarrollo realiza todas las preguntas para hacer los ajustes a sus estimaciones.

Durante el “como”, el equipo de desarrollo determina la forma en la que llevara a cabo el trabajo. Esto implica el diseño que se ira modificando durante los Sprints y la identificación de las actividades que el equipo llevara a cabo.

Alcance

Alcance Del Producto

El resultado del desarrollo producirá un producto de software llamado PreLibrerate, que permitirá a las PPL que se encuentran reclusos en el CPAMS Girón en el departamento de Santander los cuales hacen uso de la biblioteca del penal. Esto permitirá de forma optimizada en tiempo y recursos, acercar a las PPL a la biblioteca del penal e incentivar la lectura en la población privada de la libertad.

Alcance del proyecto

EL equipo de desarrollo centrará sus esfuerzos en el desarrollo de una herramienta software que permita a las PPL del CPAMS Girón acceder al inventario de la biblioteca del penal desde cualquier computador del área de educativas del penal, es decir que el PPL podrá tener acceso al inventario sin necesidad de entrar a la biblioteca la cual tiene restricciones de acceso por motivos de los niveles de seguridad.

Personal Involucrado

Tabla 4

Personal Involucrado

Nombre	Dannys Eduardo Cardozo Benítez
Rol	Analista, diseñador y programador
Categoría Profesional	Ingeniero de Sistemas
Responsabilidad	Análisis de información, diseño y programación del aplicativo web PreLibrerate
Información de contacto	dannyscardozo915@gmail.com
Nombre	Martin Camilo Cancelado
Rol	Analista, diseñador y programador
Categoría Profesional	Ingeniero de Sistemas y Especialista en Seguridad Informática
Responsabilidad	Análisis de información, diseño y programación del aplicativo web PreLibrerate
Información de contacto	martin.cancelado@unad.edu.co

Nota. En esta tabla se relacionan las personas involucradas en el desarrollo del proyecto de grado.

Tabla 5

Definiciones, Acrónimos y Abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
FTP	Protocolo de Transferencia de Archivos
JSON	Formato de intercambio de datos.
Tutor	Usuario del sistema encargado transferencia de conocimientos a estudiantes
PPL	Persona Privada de la Libertad
Prioridad	Prioridad del Requisito, esta puede ser: A=Alta/Esencial, M=Media/Deseada, B=Baja/Opcional.
Chequeo	Chequeo de verificación para cumplimiento
Tipo Requisito	Si es Requisito (RQ), si es Restricción (RS).

Nota. En esta tabla se relacionan las siglas y conceptos que utilizaremos en la definición de los requerimientos funcionales y no funcionales

Referencias**Tabla 6***Referencias*

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

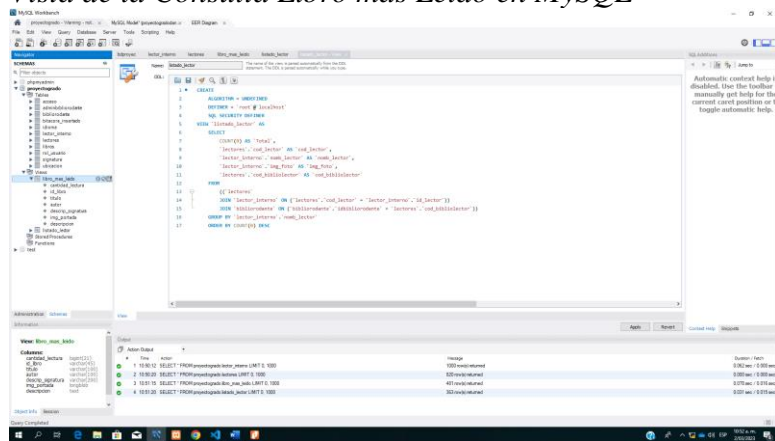
Nota. En esta tabla se relaciona el formato que utilizaremos para definir la especificación de requisitos de software según el Standard IEEE 830 - 1998

Funcionalidad del producto

Requerimiento general para PreLibrerate

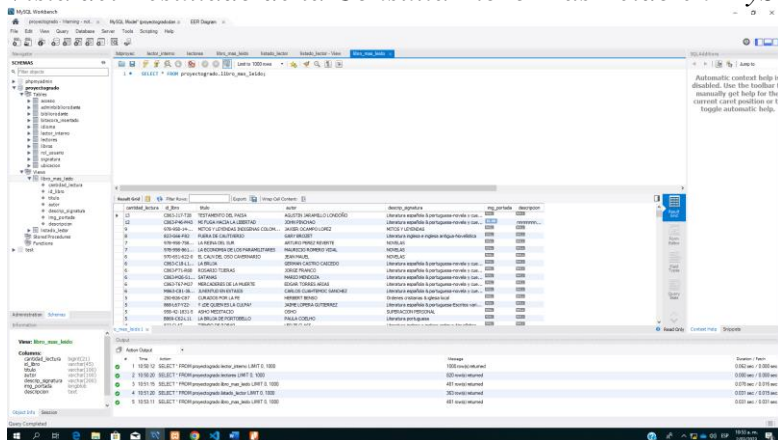
PreLibrerate permitirá que una PPL consulte el inventario de la biblioteca del penal desde cualquier equipo de cómputo del área de educativas que se encuentre conectado al servidor del área al cual están conectados aproximadamente 40 equipos para el uso de los internos, el interno del penal podrá solicitar el préstamo de un máximo de 3 libros después de haber creado el usuario, una vez validado el usuario podrá ingresar a la plataforma y consultar el material bibliográfico y solicitar su préstamo, para crear la cuenta deberá registrarse con los datos que lo identifican en el penal como son número de TD, nombre completo, patio en el cual está ubicado y como contraseña el número NUI que es el número que identifica a los internos a nivel nacional y es de conocimiento exclusivamente del INPEC y el. El aplicativo debe tener estadísticas e informes que serán consultados por el personal administrativo y de esta manera generar los préstamos solicitados y actualizar el Front End en el cual se publicara cual es el interno que más lee y el libro más leído así, información que se actualizara dependiendo el comportamiento de la actividad lectora.

Figura 4
Vista de la Consulta Libro más Leído en MySQL



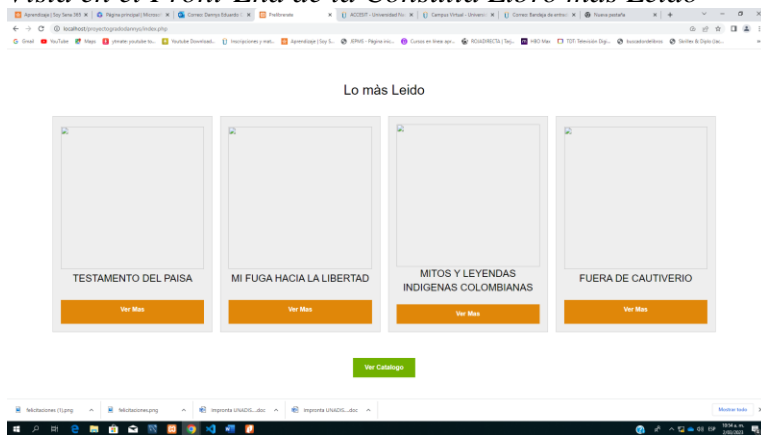
Nota. En esta imagen se puede observar la consulta libros más leídos utilizando la herramienta Mysql

Figura 5
Vista del Resultado de la Consulta Libro Mas Leído en MySQL



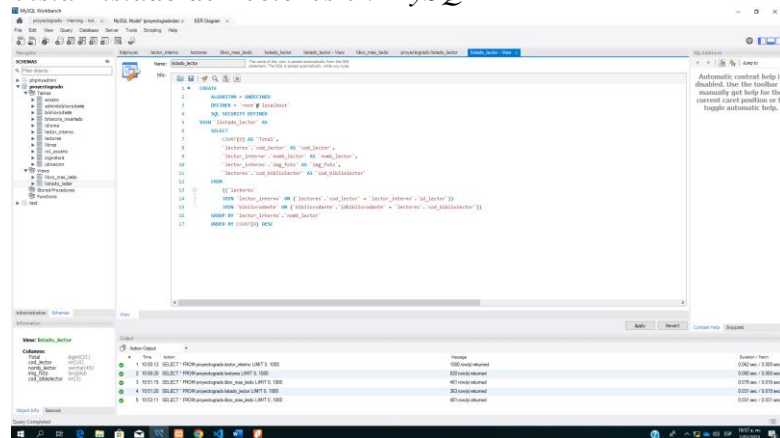
Nota. En esta imagen se puede observar el resultado de la consulta libros más leídos utilizando la herramienta Mysql

Figura 6
Vista en el Front-End de la Consulta Libro más Leído



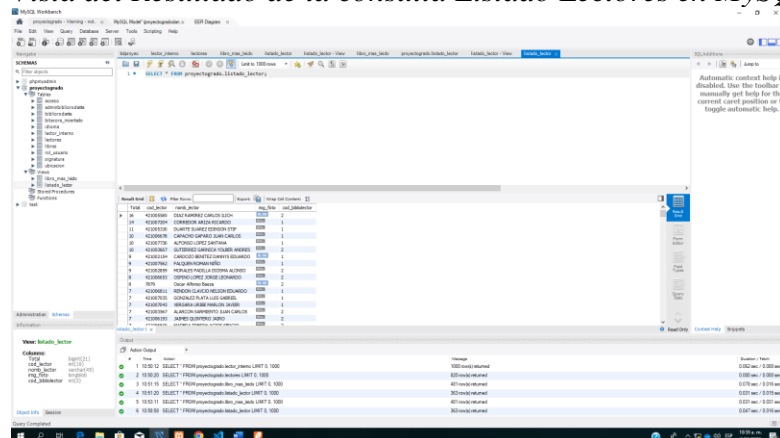
Nota. En el FrontEnd de la aplicación web se reflejan los cuatro libros más leídos por la población reclusa según la consulta realizada en Mysql.

Figura 7
Vista Listado de Lectores en MySQL



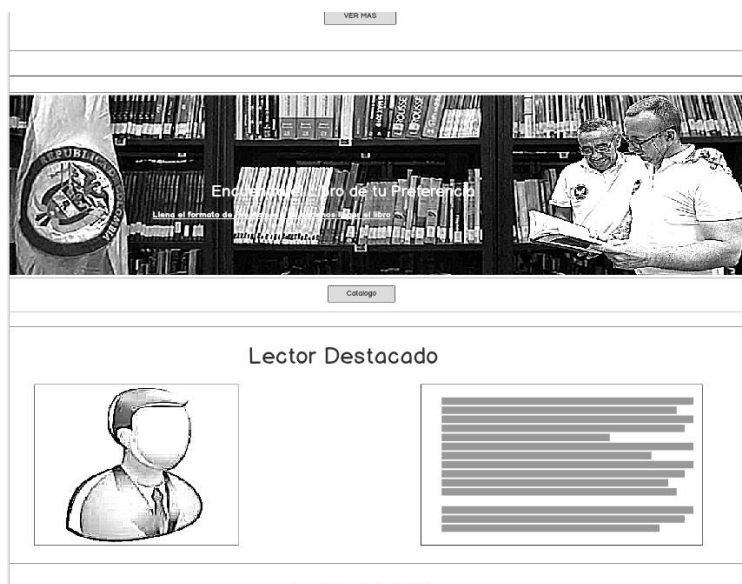
Nota. En esta imagen se puede observar la consulta listado lectores utilizando la herramienta Mysql

Figura 8
Vista del Resultado de la consulta Listado Lectores en MySQL



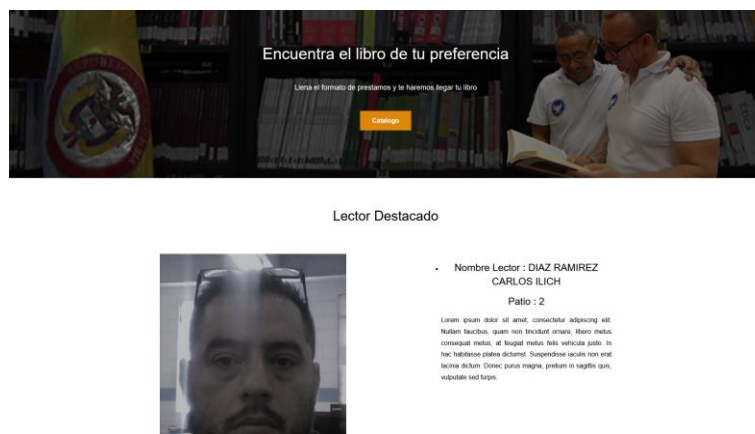
Nota. En esta imagen se puede observar el resultado de la consulta listado lectores utilizando la herramienta Mysql

Figura 9
Boceto del Front-End en la sesión lector destacado



Nota. En esta imagen podemos observar el boceto de la sesión lector destacado la cual nos ilustra cómo se vería en el navegador.

Figura 10
Vista en el Front-End en el Navegador en la Sesión Lector Destacado



Nota. En esta imagen podemos observar la sesión lector destacado la cual nos ilustra cómo se vería en el navegador.

Características de los Usuarios

Tabla 7

Usuario Administrador

Tipo de Usuario	Formación	Actividades
Administrador	Profesional en Informática	Control y Manejo del sistema en general

Nota. En esta tabla podemos observar los requisitos que debe llenar el usuario administrador.

Tabla 8

Usuario Bibliotecario

Tipo de Usuario	Formación	Actividades
Bibliotecario	PPL o funcionario con conocimientos básicos de ofimática	Control y Manejo del sistema en general

Nota. En esta tabla podemos observar los requisitos que debe llenar el usuario Bibliotecario.

Tabla 9

Usuario PPL

Tipo de Usuario	Formación	Actividades
PPL	Internos con o sin educación básica.	Participación como lectores del área de educativas y el penal en general

Nota. En esta tabla podemos observar los requisitos que debe llenar el usuario PPL.

Restricciones

- Interfaz para ser usada en ambiente web
- Uso de Dominio proporcionado por el INPEC
- Lenguajes y tecnologías que usar: HTML5, PHP7, CSS3, JAVASCRIPT.
- La aplicación se diseñará según un modelo cliente/servidor.
- La arquitectura de software de la aplicación tendrá un estilo basado en Modelo Vistas y Controladores (MVC).
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

- Base de datos MySQL.
- Para el desarrollo se usará Scrum como metodología ágil.

Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma.

Evolución previsible del sistema.

EL sistema analizará mediante las estadísticas de uso estudiando, aquellos parámetros de usabilidad por parte de las PPL Y funcionarios, así mismo se analizará la integralidad del proyecto teniendo en cuenta los resultados arrojados por el sistema para mejorar el servicio.

Requisitos Específicos

Requisitos comunes de las interfaces

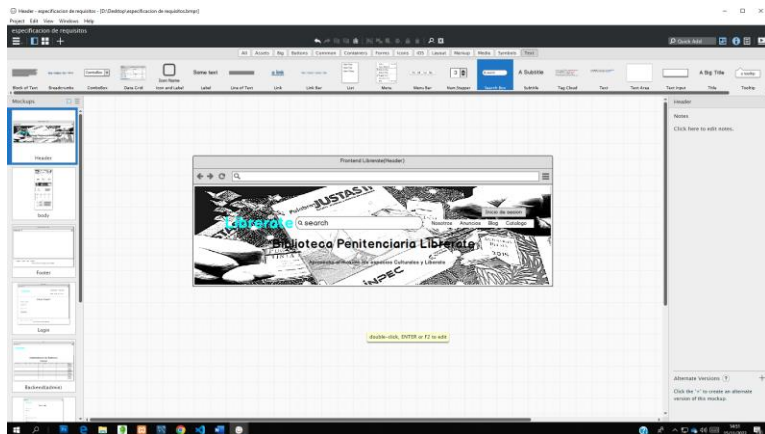
Interfaces de usuario

Generalidades. Teniendo en cuenta la integralidad con los sistemas existentes del INPEC, la interfaz gráfica debe tener en cuenta los siguientes aspectos de forma general: El cabezote y pie de página debe contemplar modelos de colores existentes en las plataformas del INPEC, se proponen los siguientes.

Cabezotes

Figura 11

Alternativa 1 de Encabezado Diseño Herramienta Balsamiq



Nota. En esta imagen podemos observar una de las opciones del encabezado o header que puede tener la aplicación web diseñado en la herramienta Balsamiq.

Figura 12 Alternativa 2 de Encabezado en Herramienta Balsamiq



Nota. En esta imagen podemos observar una de las opciones del encabezado o header que puede tener la aplicación web diseñado en la herramienta Balsamiq.

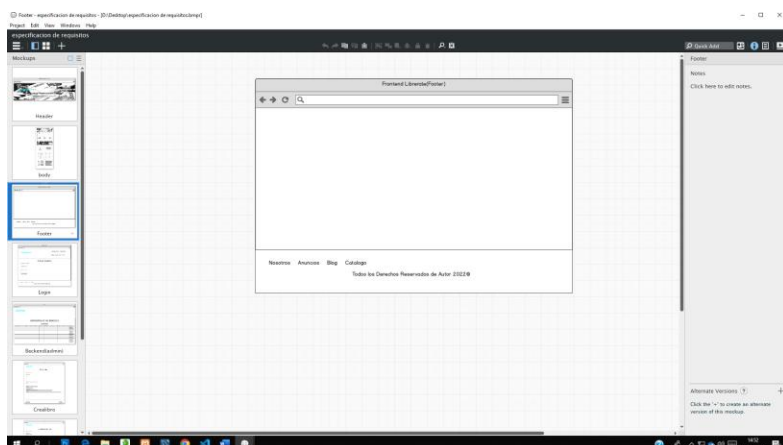
Figura 13
Alternativa de Encabezado en el Navegador



Nota. En esta imagen podemos observar una de las opciones del encabezado o header que puede tener la aplicación web vista en el navegador.

Pie de Paginas

Figura 14
Alternativa 1 Pie de Página, Diseño en Herramienta Balsamiq



Nota. En esta imagen podemos observar una de las opciones del pie de página que puede tener la aplicación web diseñado en la herramienta Balsamiq.

Figura 15 *Alternativa 2 Pie de Página en Herramienta Balsamiq*



Nota. En esta imagen podemos observar una de las opciones del pie de página que puede tener la aplicación web diseñado en la herramienta Balsamiq.

Figura 16

Alternativa 2 Pie de Página en Navegador



Nota. En esta imagen podemos observar una de las opciones del pie de página que puede tener la aplicación web vista en el navegador.

El diseño de la interfaz de usuario debe ser sensible (Responsive) a diferentes tamaños de pantalla.

El usuario podrá interactuar con el sistema haciendo uso de un navegador web y sin restricciones de tiempo.

Interfaces de hardware

El usuario PPL de PreLibrerate puede usar la solución en cualquier equipo del área de Educativas del CPAMS Girón con o sin conexión a internet.

Para la instalación del aplicativo en un servidor se requiere un equipo conectado a internet con puerto 80 abierto, memoria mínima de 8 GB teniendo en cuenta las conexiones concurrentes esperadas y almacenamiento en disco disponible de 100 MB, la escalabilidad en almacenamiento dependerá de la cantidad de registros con imágenes almacenadas.

Interfaces de software

PreLibrerate se ejecutará los sistemas operativos Windows, Mac y Linux, se precisa tener un servidor HTTP, MySQL y debe tener instalado PHP preferiblemente versión 7 o superior, adicionalmente se recomienda ser instalado en un servidor con alcance local a la base de datos de registro y control del INPEC.

Interfaces de comunicación

Para la conexión entre cliente y servidor se usará un servicio web en PHP con transferencia de paquetes de datos en JSON.

Para hacer uso de la tecnología de comunicación se usará el protocolo HTTP disponible en puerto 80.

Requerimientos Funcionales

Para facilitar el entendimiento y los procesos de los requerimientos funcionales, se han organizado en forma de listas y agrupados por segmentos de interés para el cliente y el equipo de desarrollo tecnológico, esto con el fin de optimizar la métrica.

Tabla 10

Abreviaturas Para Requerimientos Funcionales

Requerimientos Funcionales	
Convenciones	
Id:	Código del requerimiento.
Nombre.	Nombre del requerimiento.
Tipo:	Si es Requisito (RQ), si es Restricción (RS).
Pri:	Prioridad del Requisito, esta puede ser: A=Alta/Esencial, M=Media/Deseada, =Baja/Opcional.
Chk:	Chequeo de verificación para cumplimiento.

Nota. En esta tabla se relacionan las abreviaturas que se utilizaran en la definición de los requerimientos funcionales y no funcionales del proyecto de grado.

Tabla 11*Requerimientos Funcionales Para Almacenamiento*

RF.1: Almacenamiento				
ID	Nombre	Tipo	PRI.	CHK
RF.1.1		PPL existente en el penal con NUI y TD de identificación en el INPEC y con registro previo en la plataforma.	RQ	A
	Descripción	Como administrador se necesita que cada PPL tenga un registro en la plataforma con los siguientes datos: usuario (Usuario existente en el penal), nombre, NUI (Numero único de identificación del INPEC), código (TD), foto; para validar su ingreso.		
ID	Nombre	Tipo	PRI.	CHK
RF.1.2		Bibliotecarios asignados a la biblioteca del penal con registro previo en la plataforma.	RQ	A
	Descripción	Como administrador cada bibliotecario debe tener un registro en la plataforma con los siguientes datos: usuario (Usuario existente en el penal), Si es PPL (nombre, NUI (Numero único de identificación del INPEC), código (TD), foto) Si es funcionario (nombre, CC y foto); para validar su acceso.		
	Fuente	Dannys Cardozo, Consulta de datos en registros INPEC.		
ID	Nombre	Tipo	PRI.	CHK
RF.1.3		Inventario de la Biblioteca	RQ	A
	Descripción	Como administrador se necesita la información de todo el inventario de la biblioteca con la información de cada ejemplar (código, título, autor, signatura, idioma, ubicación).		
	Fuente	Dannys Cardozo, Consulta de datos en registros INPEC.		
ID	Nombre	Tipo	PRI.	CHK
RF.1.4		Almacenamiento de códigos Dewey de catalogación de libros según estándar bibliotecario francés.	RQ	A
	Descripción	Como administrador, se necesita la codificación del sistema Dewey para relacionar los libros según su género literario y ubicación del autor.		
	Fuente	Dannys Cardozo, Consulta de datos en registros sistema Dewey.		
ID	Nombre	Tipo	PRI.	CHK
RF.1.5		Almacenamiento de códigos de ubicación de la matriz de los stands.		
	Descripción	Como administrador, se necesita la codificación de la ubicación en la biblioteca según la matriz que está diseñada para este fin, la cual cataloga cada stand de la biblioteca con una numeración alfanumérica.		
	Fuente	Dannys Cardozo, Consulta de datos en registros sistema de ubicación según matriz de catalogación de los stands.		

ID	Nombre	Tipo	PRI.	CHK
	Almacenar el registro del ingreso del Bibliotecario a la plataforma.	RQ	A	
RF.1.6	Descripción	Como Administrador, se necesita conocer el registro de ingreso a la plataforma para generar informes de gestión. Con: id, fecha y hora de ingreso, fecha y hora de cierre, id del Bibliotecario.		
	Fuente	Dannys Cardozo, Consulta de datos en registro de sistema de la Biblioteca.		
ID	Nombre	Tipo	PRI.	CHK
	Almacenar el registro del ingreso de la PPL a la plataforma.	RQ	M	
RF.1.7	Descripción	Como Administrador, se necesita conocer el registro de ingreso a la plataforma para generar informes de gestión. Con: id, fecha y hora de ingreso, fecha y hora de cierre, id del PPL.		
	Fuente	Dannys Cardozo, Consulta de datos en registro de sistema de la Biblioteca.		
ID	Nombre	Tipo	PRI.	CHK
	Almacenar el préstamo del Libro.	RQ	A	
RF.1.8	Descripción	Como Bibliotecario se necesita almacenar el préstamo para el control de préstamos y estadísticas de la gestión bibliotecaria, con los siguientes datos: id del libro, id del patio, id PPL.		
	Fuente	Dannys Cardozo, Consulta de datos en registro de sistema de la Biblioteca.		

Nota. En esta tabla se definen los requerimientos funcionales para almacenamiento en el desarrollo la aplicación web.

Tabla 12

Requerimientos Funcionales Para Informes y Reportes

RF.2: Informes y Reportes.				
ID	Nombre	Tipo	PRI.	CHK
	Consulta de préstamos.	RQ	A	
RF.2.1	Descripción	Como Bibliotecario se requiere consultar los préstamos que se han realizado para hacer la entrega de estos en los patios.		
	Fuente	Dannys Cardozo.		
ID	Nombre	Tipo	PRI.	CHK
	Consulta de gestión bibliotecaria	RQ	A	
RF.2.2	Descripción	Como Bibliotecario se necesita consultar la gestión lectora para verificar el PPL que más lee y el libro que más se lee, así como el género literario más solicitado		
	Fuente	Dannys Cardozo.		
ID	Nombre	Tipo	PRI.	CHK
	Consulta de Actividad Lectora.	RQ	A	
RF.2.3	Descripción	Como Lector se necesita consultar cual es el libro más leído y el PPL que más lee.		
	Fuente	Dannys Cardozo.		

ID	Nombre	Tipo	PRI.	CHK
	Consulta de Catalogo de libros.	RQ	A	
RF.2.4	Descripción	Como Lector se necesita tener acceso a el catálogo para realizar el préstamo, así como la información de cada ejemplar.		
	Fuente	Dannys Cardozo.		

Nota. En esta tabla se definen los requerimientos funcionales para la elaboración de informes y reportes en el desarrollo la aplicación web.

Tabla 13

Requerimientos Funcionales Para Procesos

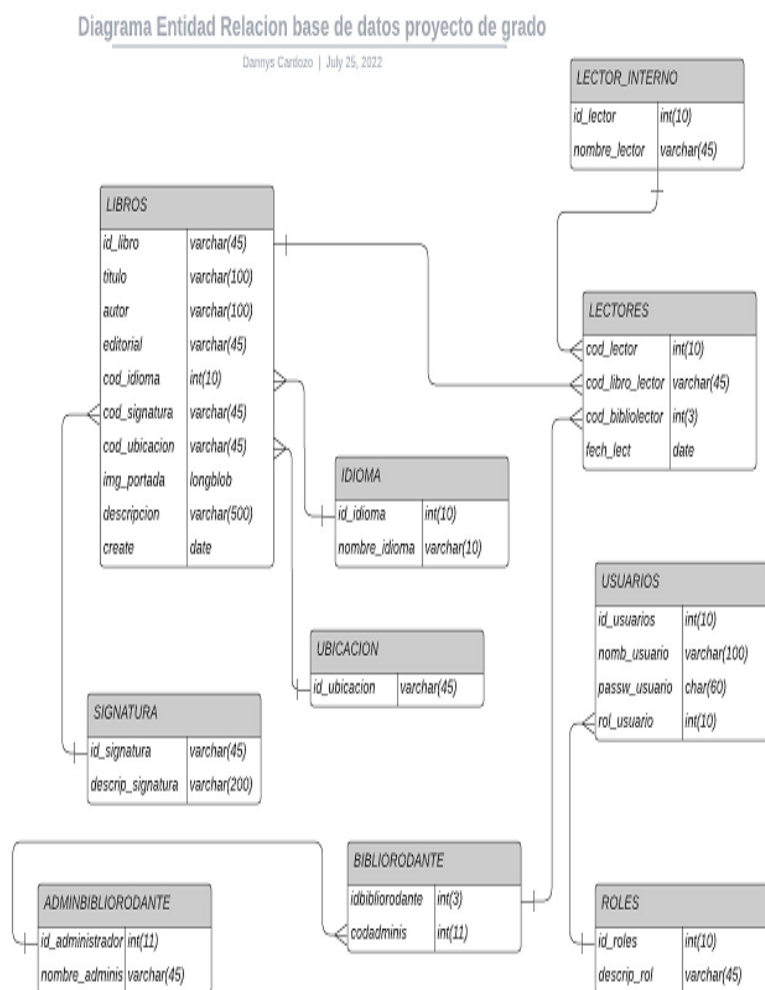
RF.3: Procesos.				
ID	Nombre	Tipo	PRI.	CHK
	Control de Prestamos por parte del bibliotecario.	RQ	A	
RF.3.1	Descripción	Como Bibliotecario se necesita poder controlar los préstamos para generar un reporte de la gestión lectora se necesita poder controlar la asistencia de los estudiantes a la tutoría para generar un registro de cumplimiento del estudiante		
	Fuente	Dannys Cardozo		
ID	Nombre	Tipo	PRI.	CHK
	Control de sanciones a PPL con determinados incumplimientos de entrega.	RQ	A	
RF.3.2	Descripción	Como administrador se necesita bloquear o sancionar temporalmente a un Lector que se demore en hacer entrega de los ejemplares prestados de forma reiterada.		
	Fuente	Dannys Cardozo.		

Nota. En esta tabla se definen los requerimientos funcionales para definir los procesos en el desarrollo la aplicación web.

Diseño

Por medio de esta fase se formula la solución de software específica, teniendo en cuenta las exigencias y necesidades, las tareas y las estrategias definidas en la fase previa, para esto el programador se encarga de diseñar la arquitectura, modela las bases de datos y el plan de diseño general, detallando los componentes concretos, las interfaces, los entornos de trabajo, entre otros componentes del software y los planes de pruebas.

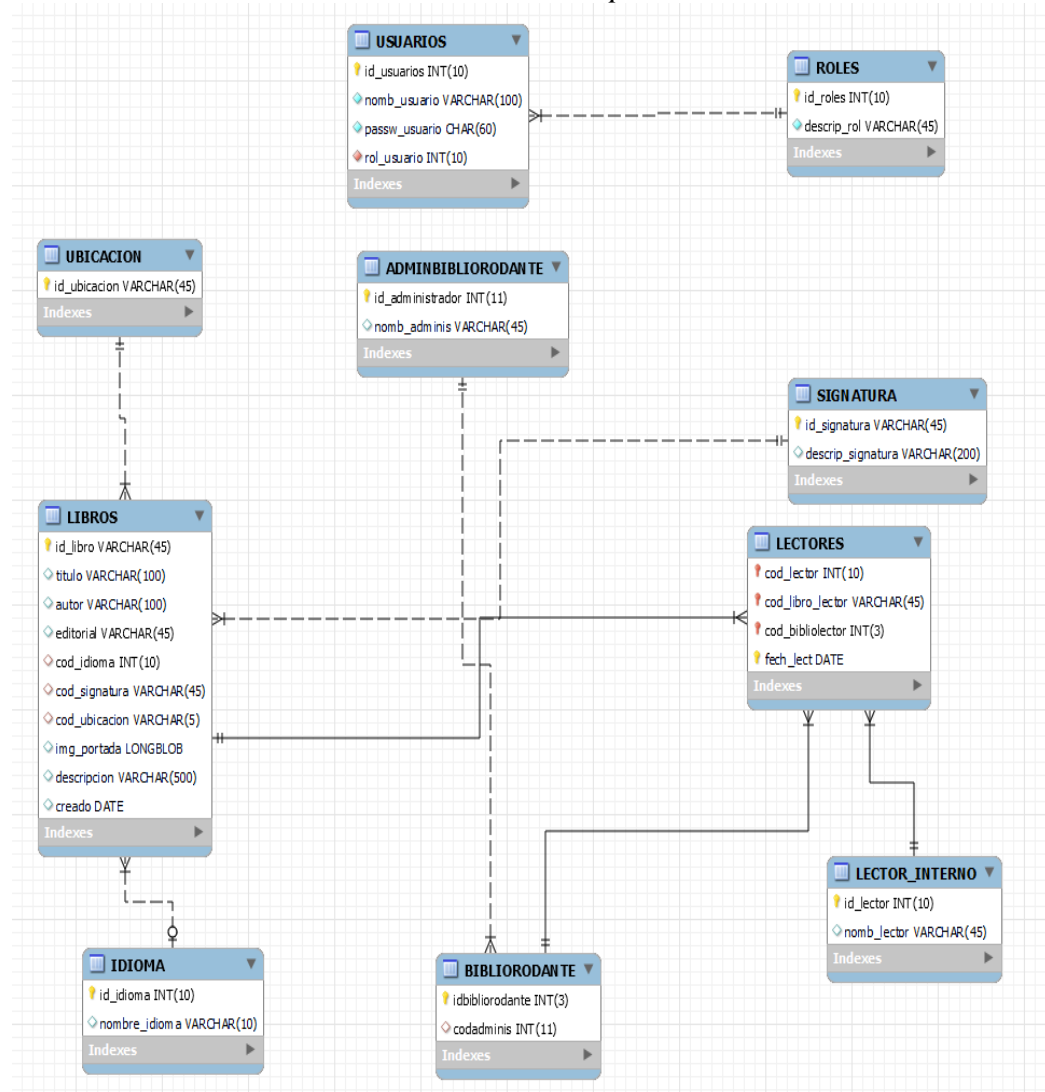
Figura 17
Modelo Lógico de la Base de Datos
Modelo Lógico DB.2022



Nota. En esta imagen podemos observar el modelo lógico de la base de datos diseñado para el desarrollo final del modelo relacional

Figura 18

*Modelo Relacional de la Base de Datos
Modelo Relacional DB. 2022.Elaboracion Propia*



Nota. En esta imagen podemos observar el modelo relacional de la base de datos utilizando la herramienta MySQL.

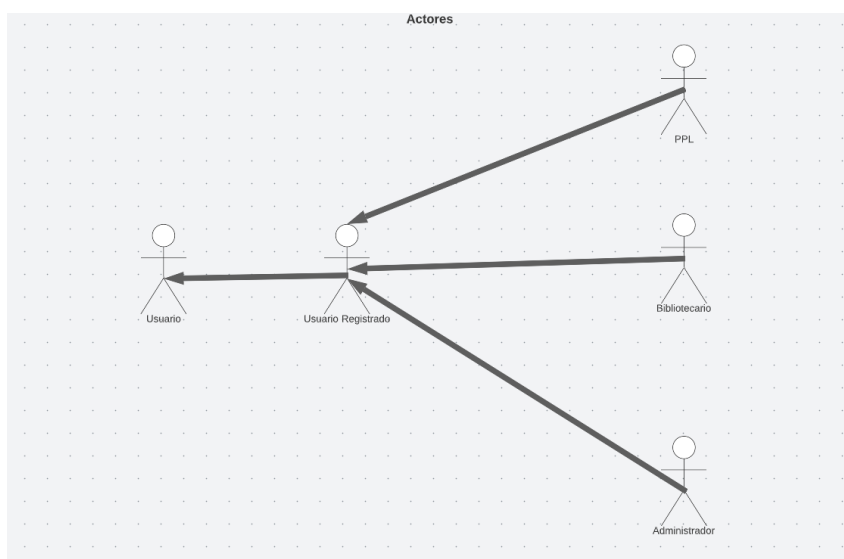
Diagrama de Casos de Uso

Actores: son los usuarios del sistema. Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.

Identificación de Actores

Figura 19

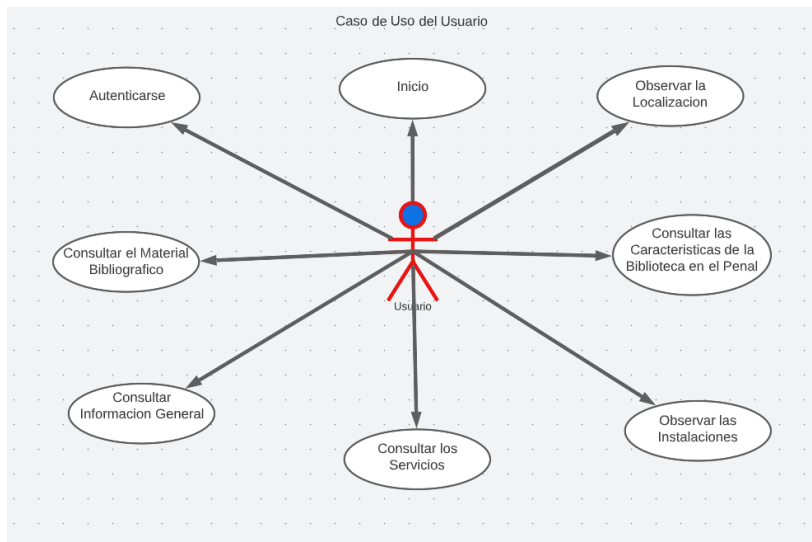
Diagrama Caso de Uso Identificación de Actores



Nota. Los usuarios son todas aquellas personas que acceden al portal web a través de internet o intranet sin necesidad de autenticarse. Los usuarios registrados son los PPL Lectores, PPL Bibliotecarios del penal y el administrador, que acceden a la aplicación mediante un nombre de usuario y contraseña.

Casos de uso del Usuario

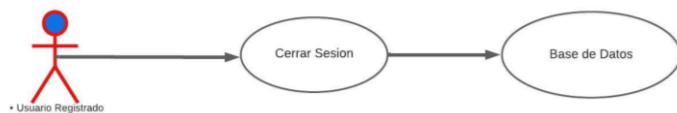
Figura 20
Diagrama Caso de Uso Usuarios



Nota. Dado que este tipo de actor, que también podría denominarse usuario anónimo o visitante, no se ha identificado ante el sistema, las acciones que podrá realizar serán muy limitadas, centrándose en la obtención de información general del centro. Sin embargo, puede acceder al inventario de la biblioteca, pero para solicitar un préstamo debe autenticarse, debe crear el usuario para acceder a una mayor funcionalidad de la aplicación, si es que dispone de un nombre de usuario y contraseña válidos.

Casos de Uso del Usuario Registrado

Figura 21
Diagrama Caso de Uso Usuario Registrado

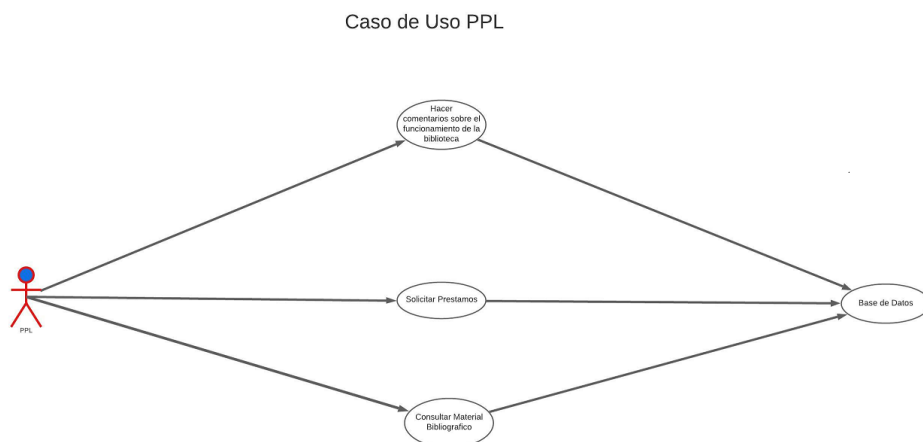


Nota. Este tipo de usuario solo puede realizar la acción de cerrar su sesión porque es la única funcionalidad que tienen en común los usuarios PPL, Bibliotecario y Administrador.

Casos de uso del PPL

Figura 22

Diagrama Caso de Uso Usuario PPL

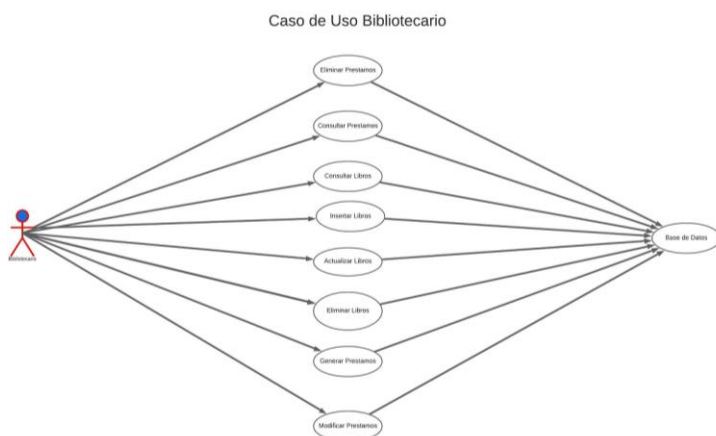


Nota. Este tipo de usuario representa a una PPL registrado en la biblioteca. Podrá realizar las acciones de consultar el material bibliográfico, solicitar préstamos y hacer comentarios sobre algún libro o funcionamiento de la biblioteca.

Casos de uso Bibliotecario

Figura 23

Diagrama Caso de Uso Usuario Bibliotecario



Nota. Este tipo de usuario representa a un bibliotecario que administra la biblioteca del penal. Podrá realizar las acciones de generar préstamos, eliminar préstamos, consultar préstamos,

modificar prestamos, insertar libros en el inventario de la biblioteca, actualizar la información de los libros en la biblioteca, consultar el material bibliográfico de la biblioteca, eliminar un libro de la biblioteca.

Casos de uso del administrador

Figura 24 Diagrama Caso de Uso Administrador

Caso de Uso Administrador



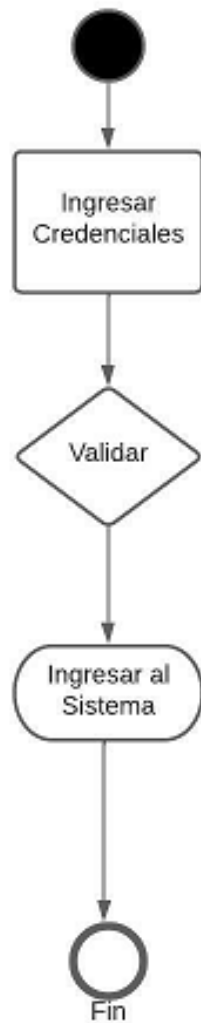
Nota. Este tipo de usuario representa al administrador de la aplicación. Podrá realizar las acciones de dar alta o baja a un usuario, dar alta o baja a un prestamos, dar alta o baja a un libro, modificar los datos de un usuario, un libro o un préstamo, consultar el historial de lectura, actualizar la información de la biblioteca y control total de la base de datos de la aplicación.

Diagramas de Actividades

Figura 25

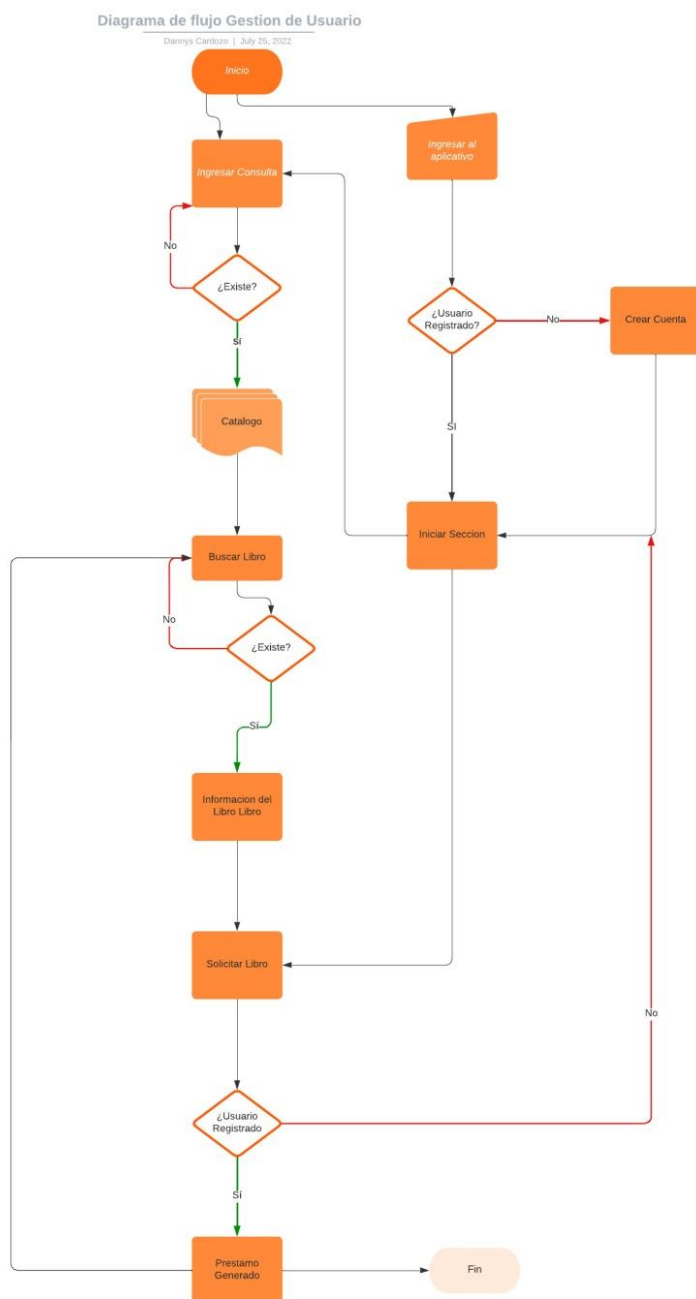
Diagrama de Actividad

Diagrama de Actividad: Autenticacion. Elaboracion Propia.



Nota. En esta imagen podemos observar el diagrama de actividad para autenticación de un usuario para poder ingresar en la aplicación web.

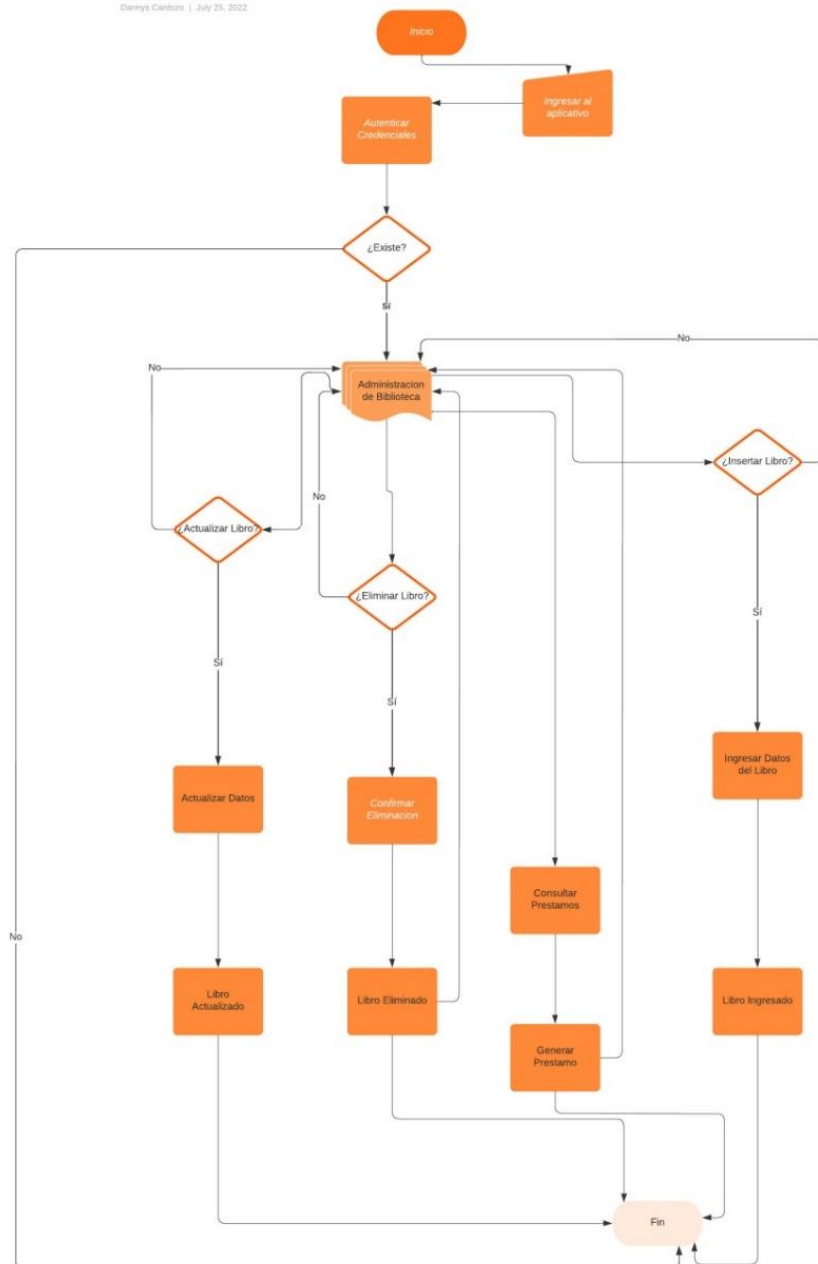
Figura 26
Diagrama de Actividad Gestión de Usuarios
Elaboración Propia



Nota. En esta imagen podemos observar el diagrama de flujo de actividades requeridas para el ingreso y gestión de préstamos de un usuario al acceder a la aplicación web.

Figura 27
Diagrama de Actividad. Insertar, Actualizar, Eliminar y Generar Préstamo
Elaboración Propia

Diagrama de flujo de Usuario Bibliotecario, Actividad
 Danyys Caribari | July 25, 2022

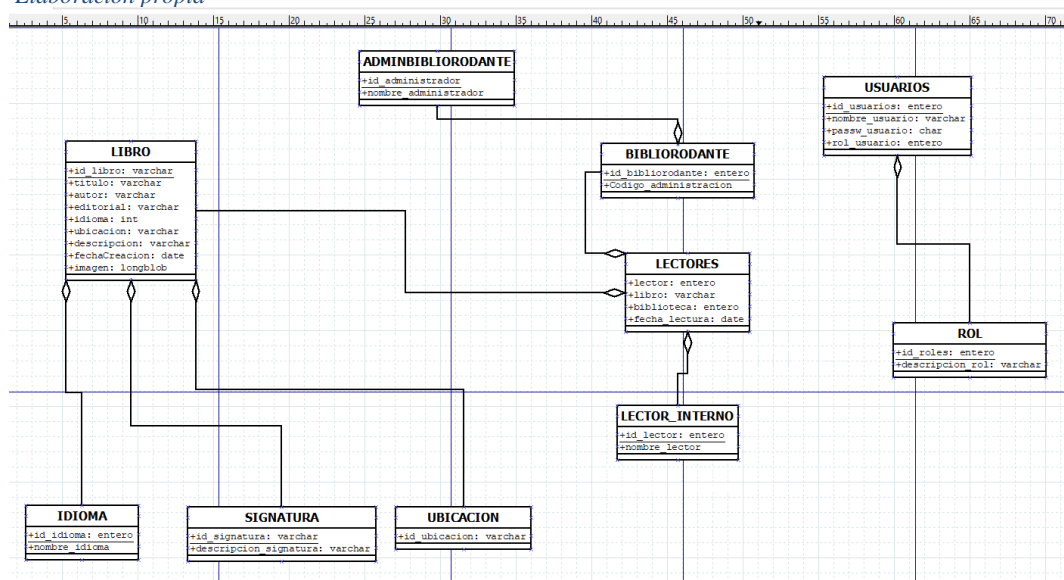


Nota. En esta imagen podemos observar el diagrama de flujo de actividades requeridas para el ingreso y gestión de inserción, actualización, eliminación y generación de préstamos de un usuario al acceder a la aplicación web.

Diagrama de Clases

Figura 28

Diagrama de Clases
Elaboración propia

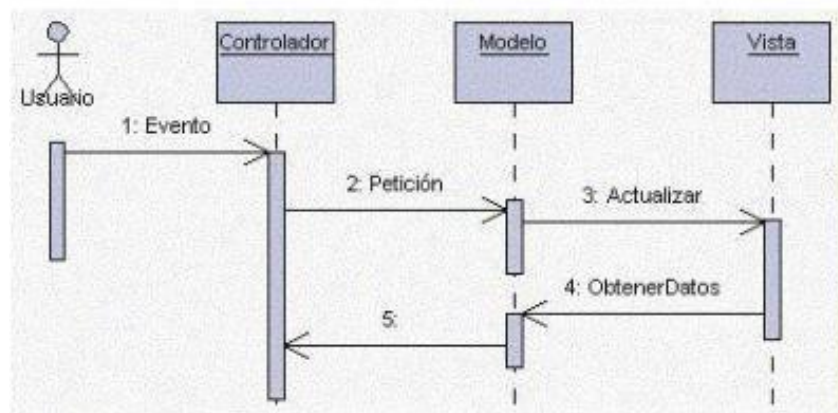


Nota. En esta imagen podemos observar el diagrama de clases elaborado en la etapa de diseño de la base de datos.

Diagrama de Secuencia

Figura 29

Diagrama de Secuencia



Nota. Patrón Modelo-Vista-Controlador. Recuperado de: Revista Digital De Tecnologías De La Información Universidad Tecnológica De La Habana.

Yanette Díaz, Yenisleidy Fernández. (2012).

Implementación

En esta fase se realiza la codificación y el desarrollo del software, se toma la información obtenida en las etapas de análisis y diseño, y esta se traduce al lenguaje de programación definido, el proyecto se divide en diversos componentes que a su vez se desarrollan por separado y se integran paulatinamente, hasta tener un producto final que se procede a probar y evaluar, para garantizar que cumpla con los requerimientos acordados.

Para realizar la implementación, inicialmente se diseña un borrador o bosquejo de la interfaz y se define la interacción del usuario con esta, para usar cada una de las herramientas definidas por el rol de cada usuario.

Diseño de la Interfaz de Usuario

Una fase fundamental para el desarrollo e implementación del Software es la planificación del diseño de la interfaz de Usuario:

Figura 30
Modulo de Registro Usuario



Nota. Vista de navegador del módulo registro de usuario desde la herramienta Balsamiq.

Figura 31
Vista Inicio de Sesión en el Navegador



Nota. Vista de navegador del módulo inicio de sesión desde la herramienta Balsamiq.

Figura 32
Vista Administración de Biblioteca en el Navegador



Nota. Vista administración de biblioteca desde la herramienta Balsamiq.

Figura 33
Diseño Registro de Libro en Herramienta Balsamiq



Nota. Módulo de crear libro, en el cual se ingresan los datos del libro

Figura 34
Vista Registro de Libro en Herramienta Balsamiq



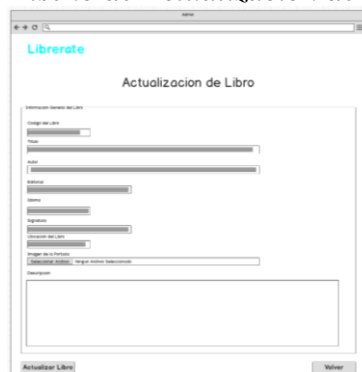
Nota. Vista registro de libro desde la herramienta Balsamiq.

Figura 35
Diseño de Eliminación de Libro en Herramienta Balsamiq

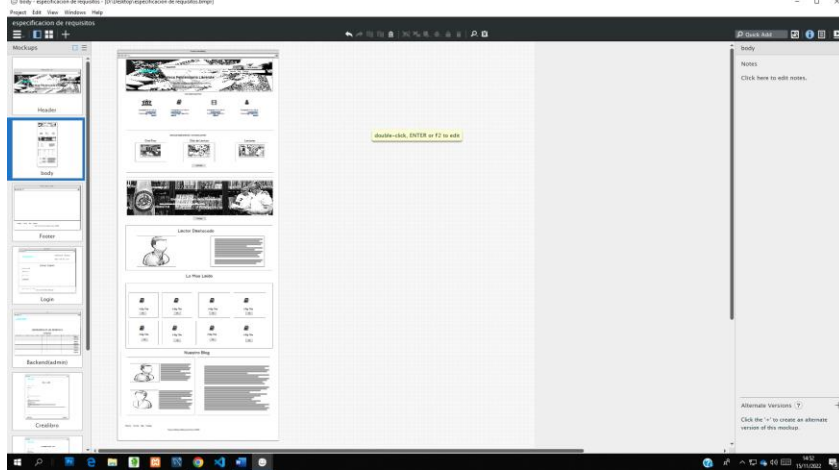


Nota. Módulo de eliminación de libro desde la herramienta Balsamiq.

Figura 36
Diseño de Actualización de Libro en Herramienta Balsamiq



Nota. Módulo de actualizar libro desde la herramienta Balsamiq.

Figura 40**Vista Completa de la Interfaz de Usuario en Herramienta Balsamiq**

Nota. Esta imagen nos muestra el boceto de la página usuario y sus diferentes sesiones desde la herramienta Balsamiq.

Figura 41**Vista del Front- End Completa en Navegador**

Nota. Esta imagen nos muestra el boceto del Front-End desde la herramienta Balsamiq.

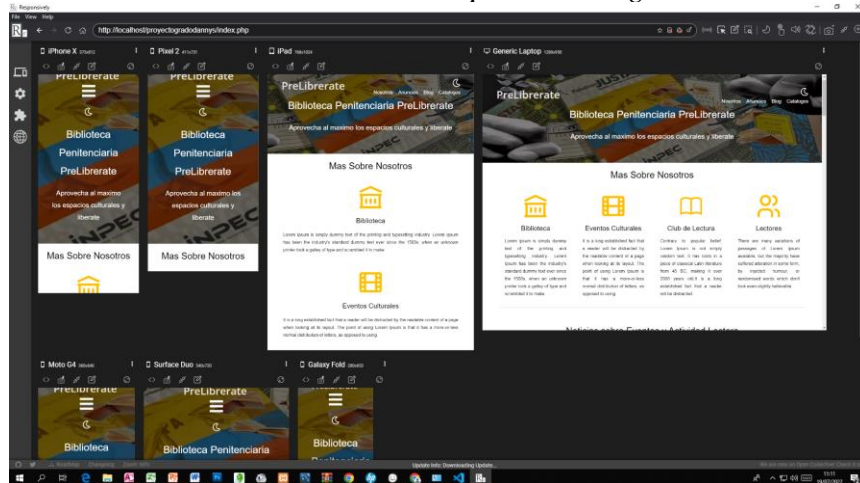
Prueba

En esta etapa del producto de software se integra con un entorno de pruebas con características similares a las que tendrían el entorno en que va a funcionar y se seleccionan algunos usuarios finales, para que utilicen el software de la manera en que requieran que funcione, evaluando que cumpla con las exigencias ya definidas, además de los estándares de calidad y funcionamiento mínimos requeridos para tener una experiencia óptima.

Software Versión Beta

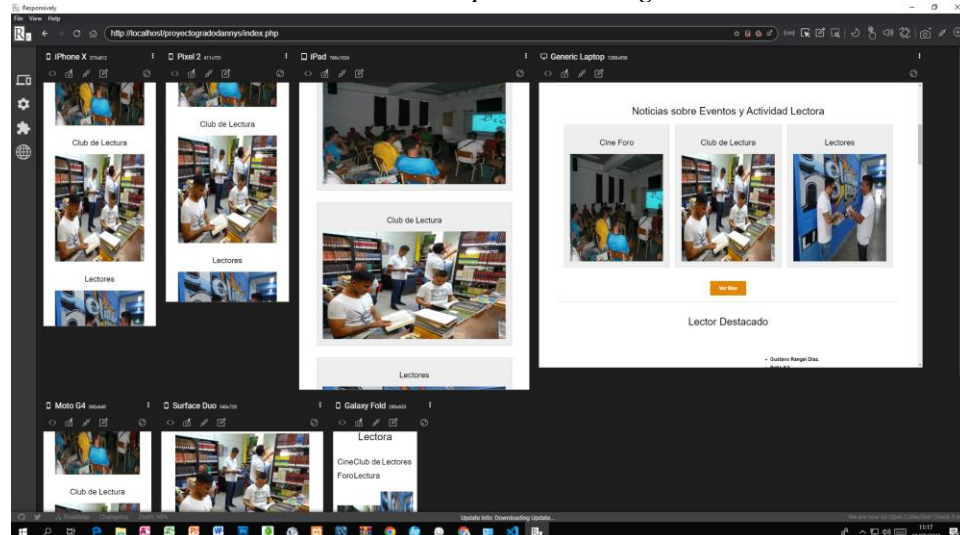
Figura 42

Vista Desde Varios Tamaños de Dispositivos Página Header



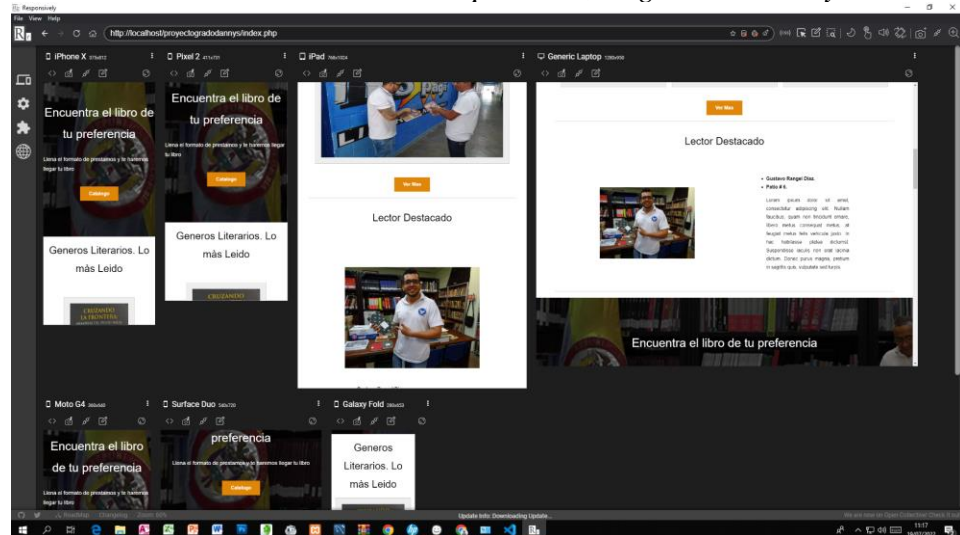
Nota. Esta imagen nos muestra el Front-End en el navegador Responsively en varios tamaños de dispositivos móviles y escritorio.

Figura 43
Vista Desde Varios Tamaños de Dispositivos, Página Header



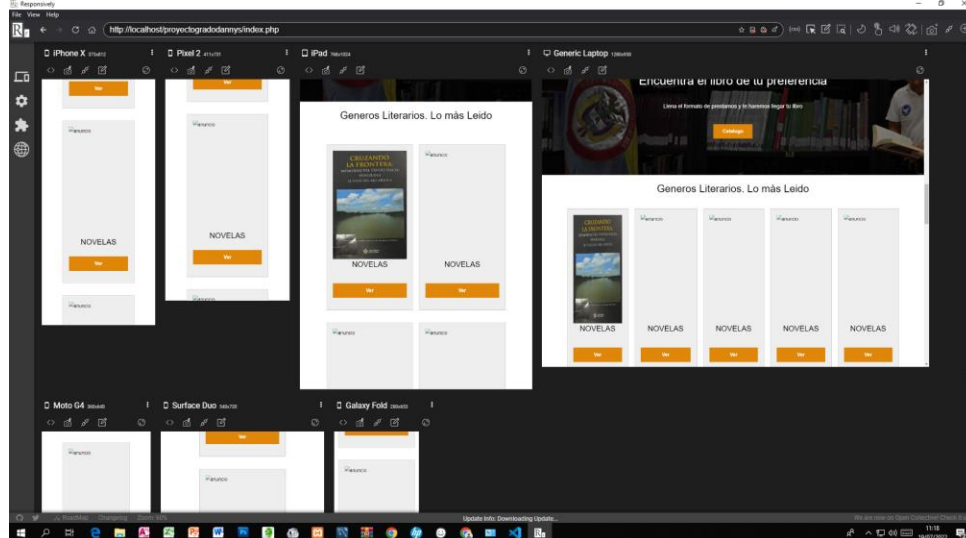
Nota. Esta imagen nos muestra el Front-End en el navegador Responsively en varios tamaños de dispositivos móviles y escritorio.

Figura 44
Vista Desde Varios Tamaños de Dispositivos, Páginas Eventos y Actividades



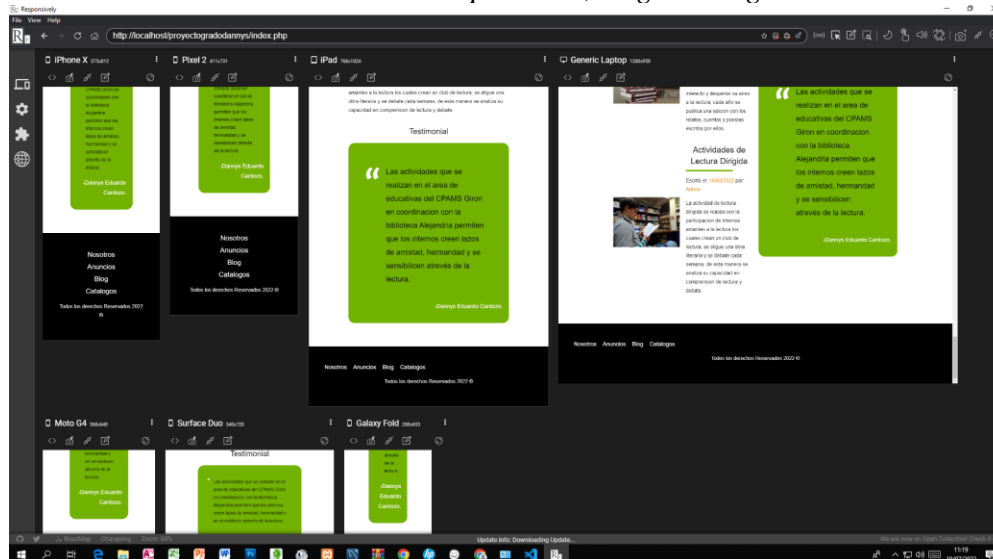
Nota. Esta imagen nos muestra el Front-End en el navegador Responsively en varios tamaños de dispositivos móviles y escritorio.

Figura 45
Vista Desde Varios Tamaños de Dispositivos, Páginas Géneros Literarios Más Leídos



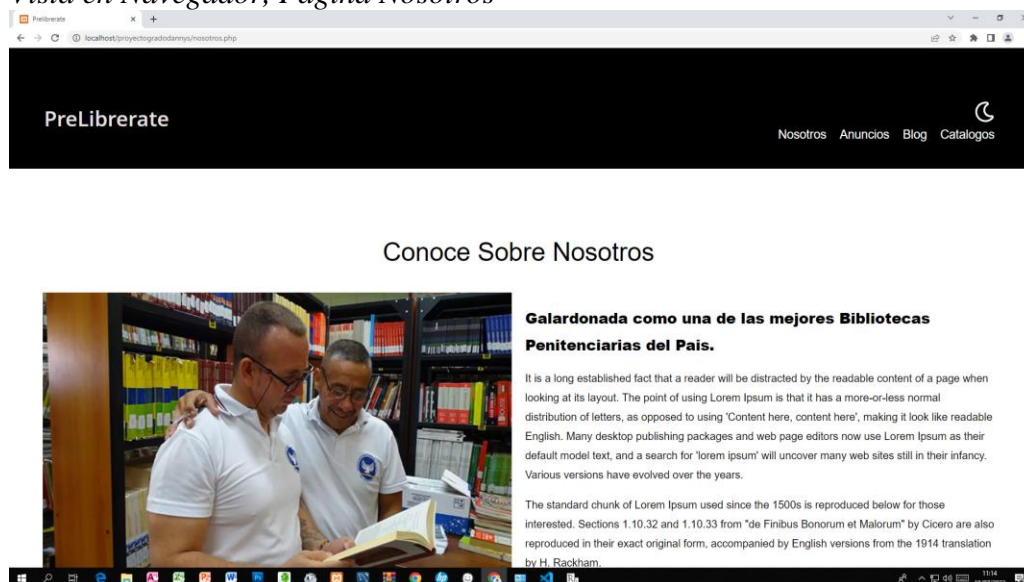
Nota. Esta imagen nos muestra el Front-End en el navegador Responsively en varios tamaños de dispositivos móviles y escritorio.

Figura 46
Vista Desde Varios Tamaños de Dispositivos, Página Blog



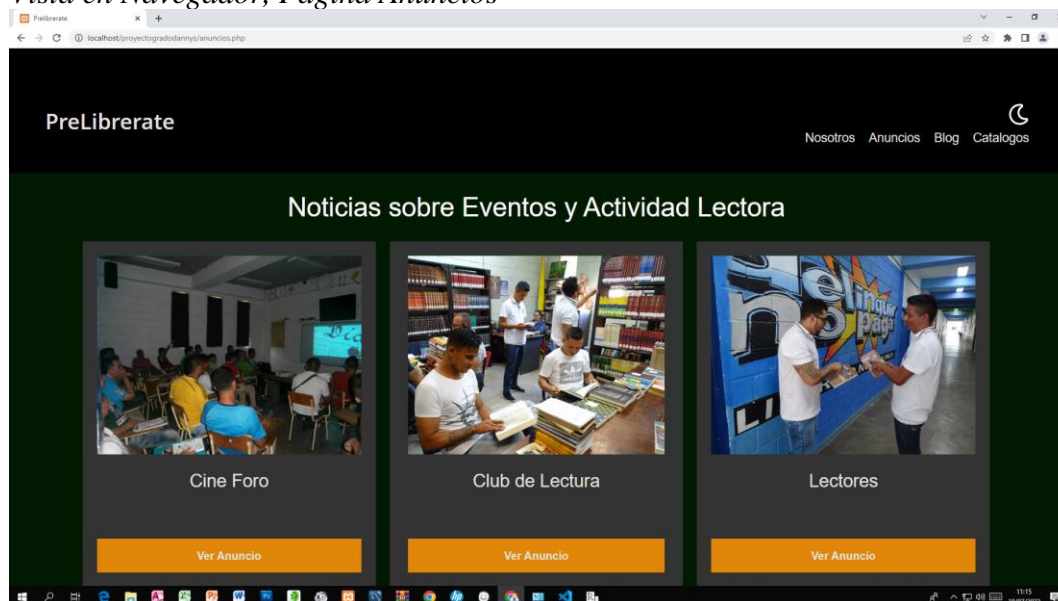
Nota. Esta imagen nos muestra el Front-End en el navegador Responsively en varios tamaños de dispositivos móviles y escritorio.

Figura 47
Vista en Navegador, Página Nosotros



Nota. Esta imagen nos la página nosotros desde el navegador.

Figura 48
Vista en Navegador, Página Anuncios



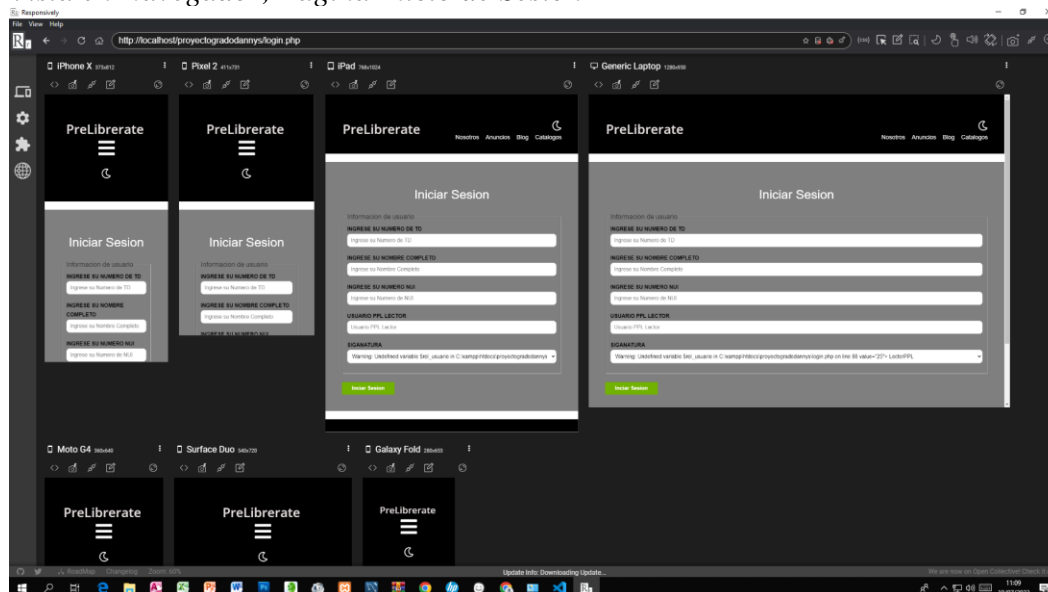
Nota. Esta imagen nos muestra la página anuncios desde el navegador.

Figura 49
Vista en Navegador, Página Blog



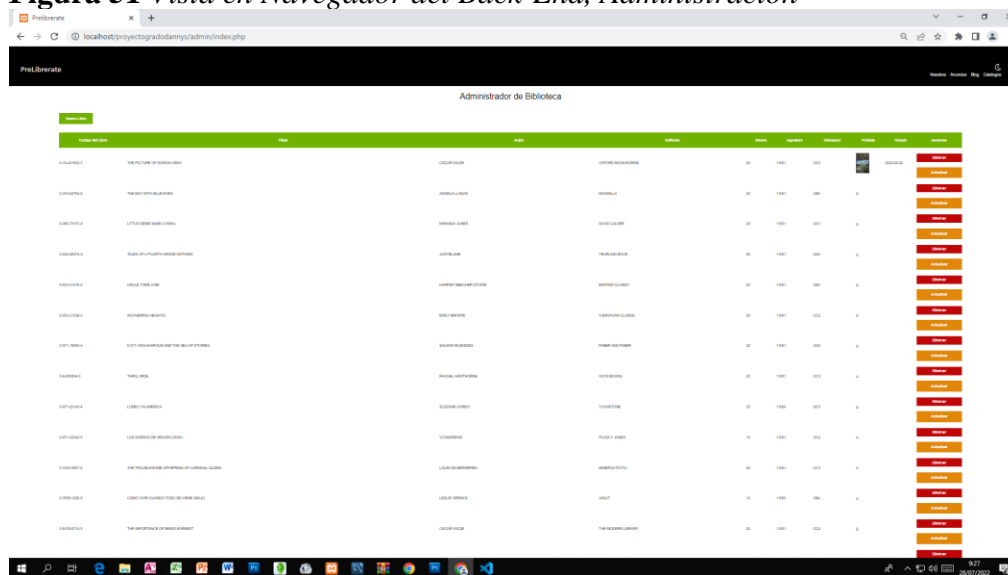
Nota. Esta imagen nos la página nuestro blog desde el navegador.

Figura 50
Vista en Navegador, Página Inicio de Sesión



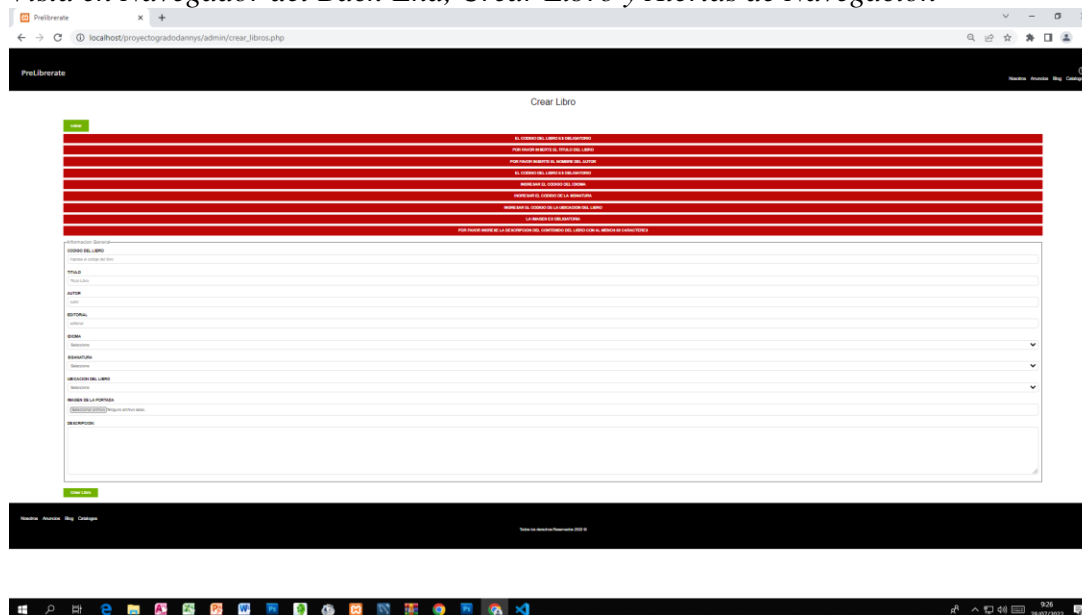
Nota. Esta imagen nos la página inicio de sesión en el navegador Responsively en varios tamaños de dispositivos móviles y escritorio.

Figura 51 Vista en Navegador del Back-End, Administración



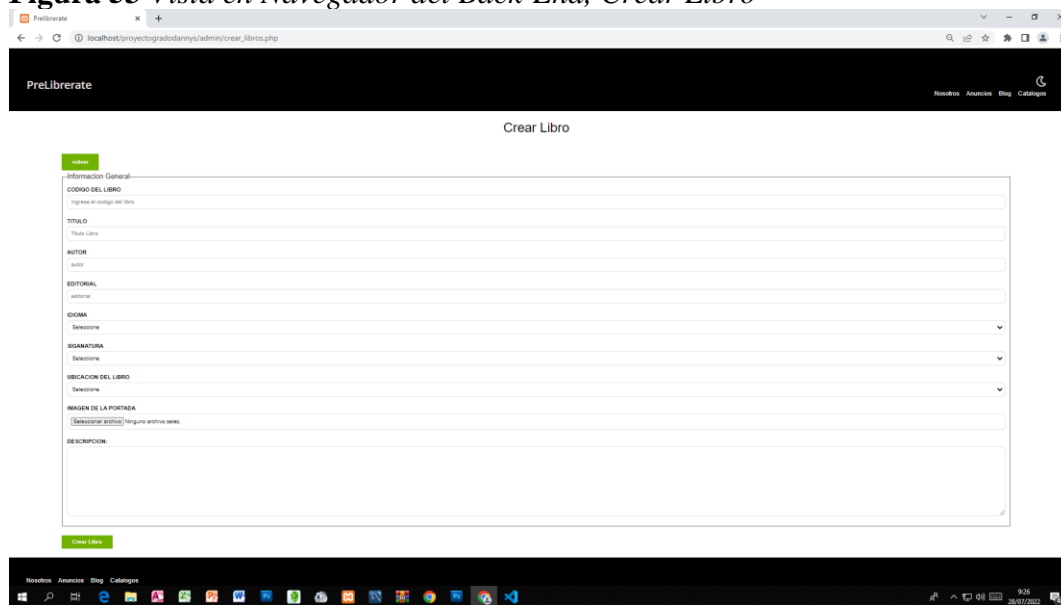
Nota. Esta imagen nos muestra el Back-End o entorno administrativo de la aplicación web en la página administración de bibliotecas.

Figura 52
Vista en Navegador del Back-End, Crear Libro y Alertas de Navegación



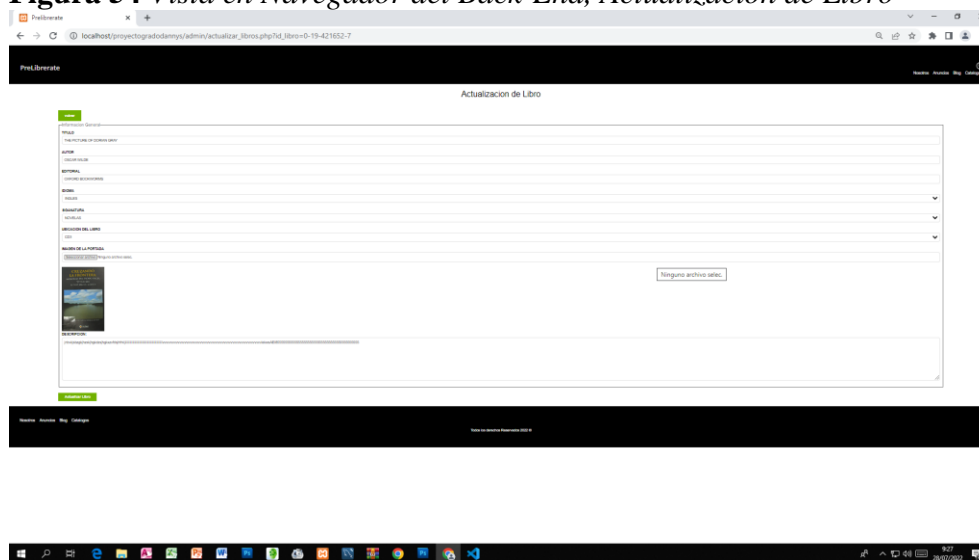
Nota. Esta imagen nos muestra el Back-End o entorno administrativo de la aplicación web en la página crear libros y alertas de navegación.

Figura 53 Vista en Navegador del Back-End, Crear Libro



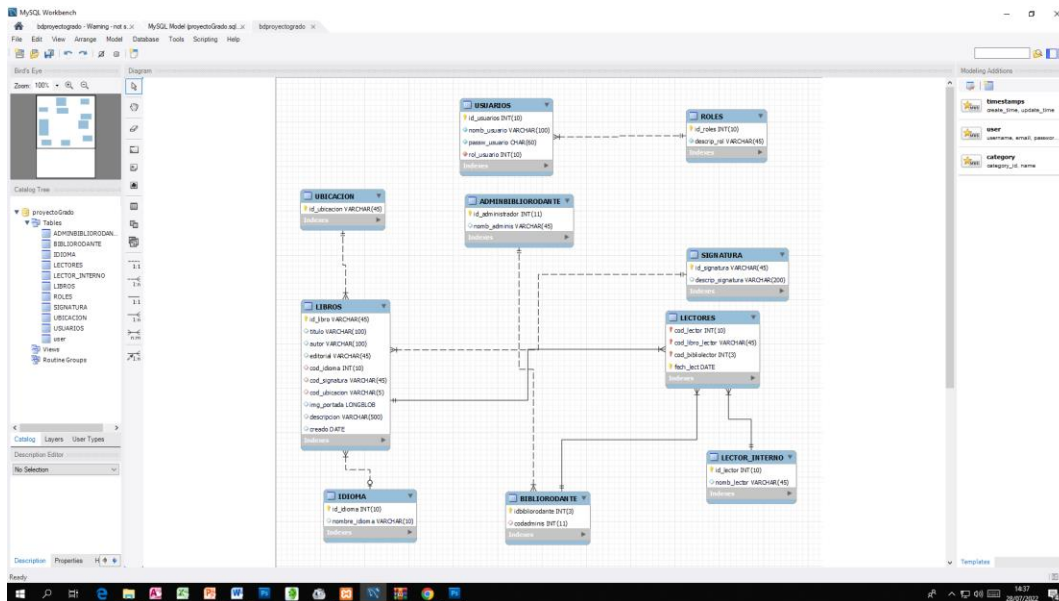
Nota. Esta imagen nos muestra el Back-End o entorno administrativo de la aplicación web en la página crear libros.

Figura 54 Vista en Navegador del Back-End, Actualización de Libro



Nota. Esta imagen nos muestra el Back-End o entorno administrativo de la aplicación web en la página actualizar libros.

Figura 55 Vista del Modelo Relacional en MySQL



Nota. Esta imagen nos muestra la estructura de modelo relacional en la herramienta *MySQL*.

Figura 56 Vista de phpMyAdmin XAMPP

The screenshot shows the phpMyAdmin interface for the 'proyectogrado' database. The main area displays a table listing the database tables. The table has columns for 'Tabla', 'Acción', 'Filas', 'Tipo', 'Cotejamiento', 'Tamaño', and 'Residuo a depurar'. The tables listed are:

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
adminbibliodante	Examinar Estructura Buscar Insertar Vaciar Eliminar	30	InnoDB	utf8_general_ci	16.0 KB	-
bibliodante	Examinar Estructura Buscar Insertar Vaciar Eliminar	13	InnoDB	utf8_general_ci	32.0 KB	-
idioma	Examinar Estructura Buscar Insertar Vaciar Eliminar	7	InnoDB	utf8_general_ci	16.0 KB	-
lectores	Examinar Estructura Buscar Insertar Vaciar Eliminar	832	InnoDB	utf8_general_ci	208.0 KB	-
lector_interno	Examinar Estructura Buscar Insertar Vaciar Eliminar	2,486	InnoDB	utf8_general_ci	160.0 KB	-
libros	Examinar Estructura Buscar Insertar Vaciar Eliminar	6,486	InnoDB	utf8_general_ci	2.2 MB	-
roles	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	16.0 KB	-
signatura	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,457	InnoDB	utf8_general_ci	96.0 KB	-
ubicacion	Examinar Estructura Buscar Insertar Vaciar Eliminar	180	InnoDB	utf8_general_ci	16.0 KB	-
user	Examinar Estructura Buscar Insertar Vaciar Eliminar	0	InnoDB	utf8_general_ci	16.0 KB	-
usuarios	Examinar Estructura Buscar Insertar Vaciar Eliminar	3	InnoDB	utf8_general_ci	32.0 KB	-
11 tablas	Número de filas	10,217	InnoDB	utf8_general_ci	2.0 MB	0 B

Nota. Esta imagen nos muestra la estructura de la base de datos en la herramienta phpMyAdmin XAMPP.

Dirección

Descripción de las fases de entrega

El proyecto se ha dividido en cinco fases o Sprints. Para cada fase se hará una entrega, para lo cual, en cada reunión de planificación del Sprint, se especificarán los puntos a entregar.

Señalado lo anterior, el proyecto se ejecutará en las siguientes fases de trabajo:

El detalle de los Sprints es el siguiente (tentativa):

Fase I. Caracterización del problema. Realizar el reconocimiento del proceso actual sobre el manejo de información en la biblioteca Alejandría.

Fase II. Análisis y Pre-diseño. Conocer el proceso desarrollado para la identificación de las variables a analizar por medio del IEEE-STD-830-1998: práctica recomendada para las especificaciones de requisitos del software.

Utilizar SCRUM como metodología ágil de desarrollo de software para facilitar el diseño de la herramienta y la aplicación de esta.

Definir el frameworks de desarrollo de software a utilizar.

Definir el diseño GUI de la herramienta y la interacción con el usuario.

Fase III. Diseño e Implementación. Diseñar la versión de prueba de la herramienta propuesta.

Fase IV. Prueba piloto. Realizar pruebas sobre las variables analizadas.

Verificar los datos generados en la herramienta diseñada.

Analizar el flujo de información del sistema.

Establecer la eficiencia de la herramienta al realizar las consultas por parte de los internos y la administración de la biblioteca Alejandría.

Fase V. Análisis y Conclusiones del modelo. Analizar la información registrada en la herramienta como una estrategia para realizar el análisis de datos para toma de decisiones.

Plantear mejoras a la herramienta diseñada con el propósito de implementarla en todas las Bibliotecas penitenciarias del país.

Para la entrega final del proyecto se tiene planificado entregar el 50% de la aplicación total.

Delimitación Temporal.

Para el desarrollo del proyecto se establece un cronograma de actividades inicial, habiendo cumplido el cronograma hasta la fase II, de la fase III en adelante se encuentra en desarrollo.

Cronograma de Desarrollo del Proyecto

Tabla 14 *Cronograma de Desarrollo del Proyecto*

Actividad	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
Fase I. Caracterización del problema.	15 de enero al 15 de marzo											
Fase II. Análisis y Pre- diseño.			16 de marzo al 16 de junio									
Fase III. Diseño e implementación.							17 de junio al 17 de septiembre					

Actividad	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7	Mes 8	Mes 9	Mes 10	Mes 11	Mes 12
Fase IV. Prueba piloto										18 de septiembre al 01 de diciembre.		
Fase V. Publicación.											02 al 15 de diciembre	

Nota. En esta tabla se refleja el tiempo estimado para el desarrollo del proyecto.

Tabla 15 *Recursos Necesarios*

Recursos	Descripción	Presupuesto
Equipo Humano	1 estudiantes y director de Ingeniería de Sistemas	1.400.000
Equipos y Software	1 equipo de mesa, 1 equipo portátil y software de desarrollo.	2.100.000
Viajes y Salidas de Campo	N/A	0
Materiales y suministros	Impresión de documentos	200.000
Bibliografía	Curso de diseño web	100.000
TOTAL:		3.800.000

Nota. En esta tabla se reflejan los recursos necesarios para desarrollar el proyecto.

Conclusion

El desarrollo de este proyecto y de este software, ha permitido aplicar diferentes tecnologías y conocimientos adquiridos en el transcurso de la carrera de ingeniería de sistemas, y de esta manera aportar a la resocialización de los internos de la cárcel de Palo gordo en el departamento de Girón Santander aportando a solucionar una problemática que se ha venido presentando durante mucho tiempo que es el acceso a el material bibliográfico de la biblioteca del penal y a tener un control en la gestión de esta actividad la cual nos permitirá tener unas estadísticas que nos permitan evidenciar cual es el gusto literario de los internos y por ende el área de tratamiento y desarrollo del penal a hacer un diagnóstico en su resocialización teniendo en cuenta factores subjetivos en la clasificación en las diferentes fases de tratamientos.

Para desarrollar este proyecto fue necesario insertar 5000 ejemplares que existían en la biblioteca y estaban registrados de forma manual en una minuta y diseñar una aplicación como plan piloto para hacer un análisis sobre la actividad lectora y si valía la pena desarrollar el proyecto y la aplicación.

Dicha herramienta fue utilizada durante seis meses y aunque solucionaba la problemática de control de lectura continuaba el problema de acceso a el material bibliográfico, de ay la idea de hacer un aplicativo web en el cual los internos pudieran acceder a los catálogos por intermedio de los equipos del área de educativas y de esta manera se soluciona el problema de acceso a el inventario total de la biblioteca del penal lo cual incentivo la actividad lectora en las PPL.

Referencias Bibliográficas

Albornoz, A. F. (2022, 18 de noviembre). *5 tipos De Paradigmas De Programación.*

CognosOnline.

<https://cognosonline.com/co/blog/que-son-paradigmas-de-programacion/>

Almeida García, Raúl. (2001) *Diseño y construcción de un prototipo de base de datos para el sistema de control de reclusos de la Dirección General de Prisioneros del Ministerio de Justicia. Bogotá.*

Atlassian. (s/f), (2022, 18 de noviembre). *Scrum.* Atlassian,

<https://www.atlassian.com/es/agile/scrum>

Caro, S. L. M. (s/f), (2022, 18 de noviembre). *Implementación de un Módulo de Calificaciones Para Instituciones de Educación.* Edu.co.

<https://repository.unad.edu.co/bitstream/handle/10596/47915/slmontoyac%20%281%29.pdf?sequence=3&isAllowed=y>

Castro D., R., (2008). *El software educativo en el entorno de los medios de enseñanza.*

<http://www.redalyc.org/pdf/1815/181518069007>

¿Cómo hacer un boceto de una página web? + EJEMPLOS. (s/f-a), (2023, 1 de marzo).

<https://www.crehana.com.>, <https://www.crehana.com/blog/estilo-vida/como-hacer-un-boceto/>

Conceptos básicos sobre bases de datos. (s/f). Microsoft.com. (2023, 6 de marzo),

<https://support.microsoft.com/es-es/office/conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>

De Bibliotecarios y Bibliotecas, F. I. de A. (s/f) (2023, 7 de marzo). *Directrices IFLA/UNESCO para el desarrollo del servicio de bibliotecas públicas.* Ifla.org. <https://www.ifla.org/wp->

[content/uploads/2019/05/assets/hq/publications/archive/the-public-library-service/pg01-s.pdf](https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1046&context=sistemas_informacion_documentacion)

De lectura, (s/f), (2023, 7 de marzo). *Ciencia Unisalle. Edu.co.*

https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1046&context=sistemas_informacion_documentacion

Definiciones: php [Wiki GUTL]. (s/f), (2023, 31 de marzo). Jovenclub.cu.

<https://gutl.jovenclub.cu/wiki/doku.php?id=definiciones:php>

Digital Marketing Specialist. (s/f), (2023, 6 de marzo). *Gestor de Base de datos: Qué es,*

Funcionalidades y Ejemplos. Entelequia. <https://intelequia.com/blog/post/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos>

Marcilla & García, (2012), (S/f),(2023, 7 de marzo). Edu.co. *Desventajas de un SGBD*

<https://repository.unad.edu.co/jspui/bitstream/10596/38212/3/srestrepom.pdf>

Edición, en su N., & Dirige, S. (s/f). (2023, 6 de marzo), *S O M M E R V I L L E*. Scalahed.com.

https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf

Encuentra aquí información de Lenguajes de manipulación y definición de datos para tu escuela

¡Entra ya! (2008, mayo 9). Rincondelvago.com.

<https://html.rincondelvago.com/lenguajes-de-manipulacion-y-definicion-de-datos.html>

Es/CSS - W3C wiki. (s/f). www.w3.org. (2023, 31 de marzo)

<https://www.w3.org/wiki/index.php?title=Es/CSS&oldid=87217>

Evolución del Software - Software. (s/f), (2022, 15 de diciembre). Google.com.,

<https://sites.google.com/site/softwarentacs/evolucion-del-software>.

García, M. (s/f), (2023, 6 de marzo). *¿QUE ES XAMPP Y COMO PUEDO USARLO?*, de <https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo/>

García, O. (2016, octubre 19). *Los tres principales roles en Scrum*. Proyectum.

<https://www.proyectum.com/sistema/blog/los-tres-principales-roles-en-scrum/>

Hardware. (2013, septiembre 21). Significados. <https://www.significados.com/hardware/>

Hojas de estilo en cascada. (2013, mayo 24). Cursos gratis.

<https://conocimientosweb.net/dcmt/ficha3807.html>

Hurtado, J. S. (2021). *¿Conoces los principios del Agile Manifestó? Thinking for Innovation*.

<https://www.iebschool.com/blog/agile-manifesto-agile-scrum/>

HTML (Historia) Rogelio Ferreira Escutia. (s/f), (2023, 7 de marzo).

Docplayer.Es., <https://docplayer.es/41870951-Html-historia-rogelio-ferreira-escutia.html>

Imbacuan, J. A. M. (s/f), (2023, 6 de marzo). *Desarrollo de un software libre educativo implementado en el área de matemáticas para grado noveno*. Edu.co., de

<https://repository.unad.edu.co/bitstream/handle/10596/13796/1126448830.pdf?sequence=1&isAllowed=y>

Importancia de la Web. (s/f), (2023, 6 de marzo). Importancia.,

<https://www.importancia.org/web.php>

IMPORTANCIA DEL DESARROLLO WEB EN EL MUNDO. (2015, noviembre

23). *Mauronet.com*. <https://www.mauronet.com/2015/11/importancia-del-desarrollo-web-en-el.html>

Ingeniería de Software, (2022, 9 de

diciembre): https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software#cite_ref-BoDu04_1-0

Installation. (s/f), (2023, 7 de marzo), Laravel.com, <https://laravel.com/docs/10.x>

Libro moderno sobre patrones de diseño: Sumérgete en los patrones de diseño, (2023, 7 de marzo) (Dakota del Norte). Refactorización.gurú, <https://refactoring.guru/es/design-patterns/book>

Lineamientos para la promoción DE lectura en la red DE bibliotecas Del Banco DE la República. (s/f), (2023, 7 de marzo) Silo.Tips, <https://silo.tips/download/lineamientos-para-la-promocion-de-lectura-en-la-red-de-bibliotecas-del-banco-de>

Llamas, J. (2020, diciembre 9). *Historia del software.* Economipedia, de <https://economipedia.com/definiciones/historia-del-software.html>.

Me. (s/f), (2023, 31 de marzo) *Educación-Financiera.* Educacion-Financiera. <https://edu.plataformal.com/blogs/Que%20Es%20Seguridad%20Informatica%20O%20Ciberseguridad>

Merchán, K. T. C. (2007). Plan estratégico para bibliotecas de centros penitenciarios de Bogotá: un acercamiento desde la biblioteca pública. *Códices*, 3(1), 75–82. <https://ciencia.lasalle.edu.co/co/vol3/iss1/6/>

Molano Gómez E. (2011). Universidad Libre, Facultad de Derecho y Ciencias Políticas de Investigaciones Socio Jurídicas, Bogotá.

Pimiento E, L.E. (2009). La Evolución del Software. Recuperado de: <http://www.monografias.com/trabajos73/evolucion-software/evolucion-software.shtml>.

protocolo http. (s/f), (2023, 31 de marzo) Uma.es. <https://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>

- Servicio de Informática. Universidad de Alicante. (n.d.)(2023, 7 de marzo) *Modelo vista controlador (MVC)*. <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Significado de Web*. (2016, junio 21). Significados. <https://www.significados.com/web/>
- (S/f). Aeemt.com. (2023, 7 de marzo),
http://www.aeemt.com/contenidos_socios/Informatica/Informac_Informat_Tecnolog/AM_V_AGI_AEEMT_HTML_Historia.pdf
- (S/f). Redalyc.org. (2023, 1 de marzo),
<https://www.redalyc.org/journal/447/44758585032/html/>
- (S/f). Www.uv.mx. (2023, 6 de marzo),
https://www.uv.mx/personal/fcastaneda/files/2015/08/F_Capitulo_5_Requerimientos_del_software.pdf
- Tatiana, K., Merchán, C., & This, A. (s/f). (2023, 7 de marzo), *Plan estratégico para bibliotecas de centros penitenciarios de Bogotá: un acercamiento desde la biblioteca pública o desde la biblioteca pública*. Edu.co.
<https://ciencia.lasalle.edu.co/cgi/viewcontent.cgi?article=1046&context=co>
- UML - Casiopea*. (s/f). Pucv.cl. (2023, 31 de marzo), <https://wiki.ead.pucv.cl/UML>
- Wikipedia contributors. (s/f-a). (2023, 7 de marzo), *Desarrollo en cascada*. Wikipedia, The Free Encyclopedia.
https://es.wikipedia.org/w/index.php?title=Desarrollo_en_cascada&oldid=140942052
- Wikipedia contributors. (s/f), (2023, 31 de marzo). *Ingeniería de software*. Wikipedia, The Free Encyclopedia.

https://es.wikipedia.org/w/index.php?title=Ingenier%C3%ADa_de_software&oldid=148633014

Wikipedia contributors. (s/f-c), (2023, 31 de marzo). *Modelo de prototipos*. Wikipedia, The Free Encyclopedia.

https://es.wikipedia.org/w/index.php?title=Modelo_de_prototipos&oldid=146619920

Wikipedia, F. (2011). *Patrones de Diseño: Patron de Diseño, Command, Singleton, Modelo Vista Controlador, Decorator, State, Lenguaje de Patron, Grasp, Bridge*. Libros LLC, Serie Wiki.

Wikipedia contributors. (s/f), (2023, 31 de marzo). *Visual Studio Code*. Wikipedia, The Free Encyclopedia.

https://es.wikipedia.org/w/index.php?title=Visual_Studio_Code&oldid=149639473

Wikipedia contributors. (s/f). *JavaScript*. Wikipedia, The Free Encyclopedia.

<https://es.wikipedia.org/w/index.php?title=JavaScript&oldid=150208690>

Xavier, F., Salgado, M., Alejandro, J., & Mena, A. (s/f), (2023, 7 de marzo). *pontificia universidad católica del ecuador facultad de ingeniería escuela de sistemas disertación previa a la obtención del título de ingeniero en sistemas y computación análisis, diseño y desarrollo de un sistema informático con interfaz web y móvil para la gestión de despacho de combustible en activos fijos y control de inventario del consumo de bodegas. caso de estudio: servicios de hidrocarburos sertecpet s.a.* Edu.ec.

<http://repositorio.puce.edu.ec/bitstream/handle/22000/19740/Trabajo%20de%20titulaci%C3%B3n%20Francisco%20March%C3%A1n%20UT2.pdf?sequence=1&isAllowed=y>