Universidad de Matanzas
Facultad de Ciencias Técnicas
Departamento Informática

TRABAJO DE DIPLOMA EN OPCIÓN DEL TÍTULO DE INGENIERO INFORMÁTICO

SISTEMA INFORMÁTICO PARA A LA GESTIÓN DE PRUEBAS FÍSICAS EN EL PROCESO DE SELECCIÓN DEPORTIVA EN LA PROVINCIA DE MATANZAS

Autor: **Javier Alejandro González Muñi**
Tutor: **M.Sc. Luis Andrés Valido Fajardo**

Matanzas, 2023

Declaración de Autoría y Nota Legal

Yo, Javier Alejandro González Muñiz , declaro que soy el único autor de la siguiente tesis, titulada Sistema informático para a la gestión de pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas y, en virtud de tal, cedo el derecho de copia de la misma a la Universidad de Matanzas, bajo la licencia Creative Commons de tipo Reconocimiento No Comercial Sin Obra Derivada, con lo cual se permite su copia y distribución por cualquier medio siempre que mantenga el reconocimiento de sus autores, no haga uso comercial de la obra y no realice ninguna modificación de ella.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____

Javier Alejandro González Muñiz

- A mi mamá, mis abuelos y mi madrina por el apoyo de estos cuatro años, se que no fueron fáciles pero estamos a un paso de lograrlo. Son una familia increible.

- A Luis Andrés Válido Fajardo, por ser el mejor tutor que podía tener, por la compresión y el apoyo desde tercer año.

- A mis amigos de A.S.E.R.E por las risas, por saber escuchar. Han formado parte de mi vida desde hace mucho tiempo, me siento muy afortunado de seguir teniendolos y compartir con ustedes este logro. Los quiero.

- A mis amigos de Los online por siempre creer en mi, por hacerme saber que si podía, por cada consejo que me hizo ser mejor. Son la prueba que la amisatd no conoce de distancia. Se los dije un día Matanzas es más linda cuando están ustedes aquí.

- A ustedes tres que serán ingenieros también, esos amigos que nos regala la carrera para hacerla más fácil. Randy te dije que tranquilo, que lo ibamos a lograr,y gracias, porque la ayuda siempre fue mutua. Leo no existe tesis en la que te pueda agradecer por todo lo que has hecho, eres una persona increíble. Ana, gracias por el apoyo, por poner el hombro en cada situacón difícil, juntos se hizo más sencillo. Sin duda los momentos con ustdes fue lo mejor que viví en la universidad.

Hay una frase que me gusta mucho y la tengo muy presente "Lo importante no es el exito, si no el camino". En el mío tuve la suerte de tener a todas estas personas exepcionales a mi lado. GRACIAS.

Resumen

En la actualidad la selección deportiva es uno de los procesos más estudiados en el mundo, la misma obliga a los estudiosos de esta rama del deporte a cada vez ser más exigentes, esto debido a la sociedad cambiante, arraigado a la evolución del hombre en el decursar del tiempo y la inclusión de la tecnología cada vez más sofisticada para conocer la manifestación del deportista en todas sus etapas de la vida. La presente investigación surge a partir de la necesidad contribuir al proceso de la gestión de pruebas físicas en el proceso la selección de atletas en la provincia de Matanzas. El objetivo fundamental es desarrollar un sistema informático que permita gestionar pruebas físicas en el proceso la selección de atletas en la provincia de Matanzas . Para el desarrollo de la solución propuesta se realiza un análisis de las principales herramientas, tecnologías y metodologías que se utilizan en la construcción de un software. El proceso estuvo guiado por el uso de las siguientes herramientas y tecnologías: *Visual Paradigm* como herramienta CASE, UML como lenguaje de modelado, Django como marco de trabajo y Python como lenguaje de programación. Finalmente se obtuvo un sistema informático que permite registrar, automatizar y generar información referente a la gestión de pruebas físicas en el proceso la selección de atletas en la provincia de Matanzas.

.

Abstract

Currently, sports selection is one of the most studied processes in the world. It compels scholars in this field to be increasingly demanding. This is due to the evolving society, intertwined with the progression of humanity throughout time, as well as the integration of progressively sophisticated technology to comprehend the athlete's performance in all stages of life. This research stems from the necessity to contribute to the management of physical tests in the athlete selection process in the province of Matanzas. The primary objective is to develop a computer system that allows for the management of physical tests in the athlete selection process in the province of Matanzas. To develop the proposed solution, an analysis of the main tools, technologies, and methodologies used in software construction is conducted. The process was guided by the usage of the following tools and technologies: Visual Paradigm as a CASE tool, UML as a modeling language, Django as a framework, and Python as a programming language. Ultimately, a computer system was created that enables the recording, automation, and generation of information concerning the management of physical tests in the athlete selection process in the province of Matanzas.

Tabla de Contenido

Introducción

En la actualidad la selección deportiva es uno de los procesos más estudiados en el mundo, la misma obliga a los estudiosos de esta rama del deporte a cada vez ser más exigentes, esto debido a la sociedad cambiante, arraigado a la evolución del hombre en el decursar del tiempo y la inclusión de la tecnología cada vez más sofisticada para conocer la manifestación del deportista en todas sus etapas de la vida.

En consideración a lo antes referido nos percatamos que cada vez se hace necesario contar con un sistema de selección correctamente avalado, permite que en una población determinada se seleccionen los mejores talentos, lo cual deviene no sólo en mayores resultados sino también en una óptima utilización de los recursos materiales, económicos, técnicos y también humanos.

El déficit de recursos materiales deportivos, los problemas existentes de infraestructura: organización, carencia de áreas deportivas, desvinculos de las entidades que intervienen en el proceso. Todo esto atenta negativamente en el buen desempeño de cualquier deporte.En los últimos años los trabajos científicos sobre la selección deportiva han tomado un auge a nivel internacional, esto propiciado por la evolución constante de los seres humanos y por la utilización cada vez mas afanada de los medios de informatización. Cada trabajo que se realiza en aras de mejorar la calidad en la selección de nuestros deportistas no es suficiente, debido a la sociedad cambiante y a las demandas que exige cada uno de los deportes en la actualidad.

La gestión en la selección deportiva en la provincia Matanzas se realiza en correspondencia con las orientaciones de la dirección nacional del Instituto Nacional de Deporte y Recrea-

ción (INDER).

El primer paso del proceso de selección deportiva comienza la realización de pruebas de eficiencia física al individuo bien en Educación Física o los Combinados Deportivos . De los resultados arrojados por el individuo en dichas pruebas del análisis posterior de dichos resultados se valora si el individuo puede ser o no un talento deportivo. Este primer eslabón presenta algunos problemas como son:

1. Al individuo no siempre se realizan todas las posibles pruebas de eficiencia de física, por varias razones:
    - Desconocimiento de los encargados de su realización.
    - No son del interés de los encargados de su realización porque dichas pruebas responde a deportes que no son del ínteres de ellos o no son del arraigo popular dentro del territorio lo que provoca la eliminación de posibles talentos.
2. No existe un sistema donde se pueda registrar estos resultados que permita ser consultado por otros especialistas que den una segunda valoración a dichos resultados.
3. No existe el mecanismo que permita comparar dichos resultados con resultados de individuos ya considerados talentos deportivos en similar etapa de la vida.

Teniendo en cuenta la situación planteada con anterioridad se define el siguiente **problema de investigación**: ¿Cómo desarrollar un sistema informático o software que contribuya a la gestión de pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas.?
Planteándose como **hipótesis**: Mediante el uso de las herramientas y tecnologías actuales, es posible desarrollar un sistema que contribuya a la gestión de pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas.
Para dar solución a este problema se asume como **objeto de estudio**: La gestión de pruebas físicas en el proceso de selección deportiva en Cuba.

En concordancia con lo anterior se propone como **objetivo general**: Desarrollar un sistema informático que permita aumentar los grados de la gestión de pruebas físicas en el proceso de selección deportiva en la Provincia de Matanzas.
Enmarcado en el **campo de acción**: La gestión de pruebas físicas en el proceso de selección deportiva en la Provincia de Matanzas.

Posibles resultados:

1. Sistema para registrar, automatizar y generar información al proceso de gestión de pruebas físicas de la selección deportiva en la provincia de Matanzas.

2. Mecanismo para la generación digital y formato duro del estado de la gestión de pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas.

3. Herramienta de apoyo para el análisis y toma decisiones de los directivos, especialistas y entrenadores que participan en la gestión de pruebas físicas del proceso de selección deportiva en la provincia de Matanzas.

Para dar cumplimiento a los objetivos de esta investigación se definieron las siguientes **tareas investigativas**:

1. Elaboración del marco teórico de la investigación a través del estudio del estado del arte que existe actualmente sobre el tema.

2. Identificación de los principales elementos que componen las pruebas físicas en el proceso de selección deportiva en Cuba.

3. Identificación de los principales elementos que componen las pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas.

4. Caracterización de los principales elementos que componen las pruebas físicas en el proceso de selección deportiva en Cuba.

5. Caracterización de los principales elementos que componen las pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas.

6. Realizar el levantamiento de requisitos funcionales y no funcionales.

7. Implementar el sistema para brindar solución al problema planteado.

8. Realizar las pruebas para validar el cumplimiento de los requerimientos.

Durante la investigación se llevan a cabo varios métodos y técnicas en la búsqueda y procesamiento de la información como son: Métodos Teóricos

- **Inductivo-Deductivo:** Para realizar el estudio de las principales herramientas existentes para el desarrollo de los sistemas informáticos para la gestión de pruebas físicas y según las características de las mismas, se definieron las cualidades que debe cumplir el sistema que se propone en el presente investigacion de diploma.

- **Analítico-sintético:** Para el estudio de los conceptos vinculados en los sistemas informáticos para la gestión de pruebas físicas, y para el análisis de la documentación necesaria, permitiendo así, un mejor entendimiento del problema a resolver y realizar la extracción de los elementos más importantes para el desarrollo del trabajo.

- **Histórico-lógico:** Para realizar un análisis de las soluciones similares y las tendencias actuales en los sistemas informáticos enfocados en la gestión de pruebas físicas.

- **Histórico-Lógico:** Para realizar el estudio de las principales herramientas existentes para el desarrollo de los sistemas informáticos para la gestión de pruebas físicas y según las características de las mismas, se definieron las cualidades que debe cumplir el sistema que se propone en el presente trabajo de diploma.

- **Modelación:** Para representar gráficamente conceptos y procesos con la finalidad de un mejor entendimiento de la solución que se propone.

Métodos Empíricos

- **Observación científica:** Para conocer el funcionamiento actual del proceso de gestión de pruebas físicas en el proceso de selección deportiva en la provincia de Matanzas, lo que permitió detectar las dificultades existentes en dicho proceso.

- **Consulta bibliográfica:** Para consultar y analizar las fuentes de información relacionadas con los diversos tipos de sistemas informáticos para la gestión de pruebas físicas.

- **Generalización:** Permite sistematizar en cada capítulo de la investigación los aspectos más significativos y llegar a conclusiones más objetivas y explícitas.

Técnicas para la obtención de información:

- **La entrevista:** Para obtener información de como se desarrolla el proceso de selección de atletas en la provincia de Matanzas, se acudió a los Combinado Deportivos Lázaro Cordoví y Napoleón Heredia ambos en el municipio de Unión de Reyes, se tomaron las notas necesarias sobre el proceso de selección de altetas.

La estructura del documento se resume en los siguientes acápites:

Capítulo 1. Fundamentación teórica: En el presente capítulo se exponen conceptos fundamentales asociados a la selcción de atletas en la provincia de Matanzas. De igual manera se analizan aplicaciones de gestión de pruebas físcas para la selección de atletas. Además se exponen las características principales de estos sistemas. Se analizan las principales tendencias, tecnologías, metodologías y softwares utilizados en la actualidad para el desarrollo de aplicaciones de gestión . A su vez se analizan y se fundamenta la selección de estas para el desarrollo de la solución propuesta.

Capítulo 2. Análisis y diseño del sistema: Se reflejan las actividades realizadas en los procesos de análisis y diseño de la solución propuesta; proceso que será guiado por la metodología de desarrollo seleccionada. En el mismo se realiza el modelo de dominio donde se describen las entidades que intervienen con el objetivo de facilitar la comprensión de los principales conceptos que se utilizarán en el proceso de negocio identificado. Se exponen los artefactos más importantes que describen el flujo normal de eventos que ocurren en el sistema, se realiza una descripción de la solución propuesta, planteándose los requisitos funcionales y no funcionales. Se define la arquitectura que tendrá la solución propuesta, asi como se definen las pruebas a realizar.

Capítulo 3. Implementación y pruebas: En este capítulo. Se realizan además una serie de pruebas que permiten validar el correcto funcionamiento de la solución, verificándose así, que el mismo cumple con todos los requerimientos y exigencias del cliente. Se muestran algunas de las vistas del sistema.

Finalmente, se presentan las conclusiones y las recomendaciones de la investigación para dejar el camino abierto a futuros estudios relacionados con el tema abordado. De igual forma, quedan recogidas las bibliografías y anexos que fueron utilizados y conformados respectivamente para el desarrollo de la solución.

Capítulo 1: Fundamentación Teórica

En el presente capítulo se exponen conceptos fundamentales asociados a la selcción de atletas en la provincia de Matanzas. De igual manera se analizan aplicaciones de gestión de pruebas físcas para la selección de atletas. Además se exponen las características principales de estos sistemas. Se analizan las principales tendencias, tecnologías, metodologías y softwares utilizados en la actualidad para el desarrollo de aplicaciones de gestión . A su vez se analizan y se fundamenta la selección de estas para el desarrollo de la solución propuesta.

1.1 Selección de atletas en la provincia de Matanzas

El mundo deportivo ha cambiado, igual que han cambiado las actitudes y el comportamiento que tiene la sociedad respecto al deporte. La gestión deportiva siendo un ámbito de este mismo fenómeno es relativamente reciente, encontrándose un aumento en su producción científica aportando investigaciones que ayudan a mejorar el proceso de toma de decisiones de los gestores del deporte. Las organizaciones deportivas siempre han tenido demanda para consumir la oferta que generaban, pero el ocio, Turismo Deportivo y la práctica de la actividad física y deportiva son valores en alza dentro del mercado, y con seguridad en los próximos años, la iniciativa va a realizar sustanciosas inversiones, esto les va a imponer el cambio en su concepción y diseño.

El Proceso de Gestión en las organizaciones deportivas de base en Cuba, ha transitado por un camino de perfeccionamiento y adecuaciones de las realidades cambiantes ligado al propio proceso revolucionario con todas sus transformaciones, en virtud del acelerado desarrollo social experimentado por la sociedad cubana, donde el deporte en todas sus

manifestaciones se elige como uno de los líderes de este proceso.

En la actualidad hablar de la organización deportiva, exige realizar un esfuerzo para integrar las múltiples y variadas tendencias que se están produciendo en su ámbito de desarrollo. Como en el mercado, la cultura, el empleo, la empresa, el hombre, etc., en el deporte se manifiestan los cambios con mayor celeridad y nitidez que en otras áreas debido, fundamentalmente, a dos cuestiones; primeramente a su juventud (un siglo en su concepción moderna), la segunda, por su consideración de fenómeno social, haciéndose eco en los medios y canales o vías de comunicación de más impacto social y la tercera, por la posibilidad que tienen nuestros deportistas de ser contratados en el exterior ya sea en becas ofertadas, o clubes profesionales que están insertados en las primeras, segundas y terceras divisiones del deporte en el mundo, ofreciéndoles técnicas, métodos y herramientas novedosas de las cuales prescindimos en nuestro país. De ahí que se hace cada vez más necesario de la selección oportuna de nuestros deportistas desde la base para su arribo al alto rendimiento, por lo que se hace necesario mejorar el funcionamiento del proceso.

Para la selección de futuros atletas en Cuba juega un papel primordial la Educación Física y los Combinados Deportivo, tienen una doble responsabilidad: educar físicamente a la población infantil y juvenil en general y lograr que los niños y las niñas entiendan que la actividad física y el deporte son para toda la vida, con el fin de prevenir la enfermedad, mejorar la calidad de vida, integrarse a la comunidad, otros y además tienen la tarea de ayudar a detectar el talento de los niños y orientarlos hacia la práctica sistemática del deporte a través de las instituciones especializadas del estado o de los organismos de deporte asociado y bajo la dirección de entrenadores especializados.

La gestión en la selección deportiva en la provincia Matanzas se realiza en correspondencia con las orientaciones de la dirección nacional del INDER. Se organiza y ejecuta mediante indicaciones técnicas, metodológicas y organizativas que se trasmiten a las estructuras funcionales que atienden esta actividad en los centros provinciales de alto rendimiento y direcciones municipales de deportes con sus correspondientes combinados deportivos, me-

diante seminarios que se realizan, como actividad preparatoria, antes del inicio de cada curso escolar. Es regulada por el Departamento de Actividad Deportiva, que cuenta con los siguientes grupos de trabajo en su estructura:

- Proyección, control y evaluación de la preparación estratégica.
- Iniciación y desarrollo del deporte en la comunidad.
- Control y evaluación de las reservas deportivas.
- Trabajo de formación integral, valores y trabajo político e ideológico.

Además del departamento y sus grupos de trabajo, existen otras organizaciones que intervienen de manera directa en la gestión y ejecución de las actividades que se realizan en el Deporte de Alto Rendimiento en Matanzas como son:

- Las comisiones deportivas provinciales.
- Las direcciones municipales de deportes.
- Las comisiones deportivas municipales.
- Las escuelas provinciales de formación de atletas.
- Los combinados deportivos.

Como parte del control y evaluación del cumplimiento de la estrategia, los directivos de la provincia de Matanzas realizan un análisis de los resultados de todos los años correspondientes al presente ciclo olímpico que permite conocer la existencia de insuficiencias en la sistematización de las actividades que se desarrollan en la Base y en el Deporte de Alto Rendimiento, donde no se logra una adecuada integración entre los diferentes niveles de actuación; escuela y área deportiva, combinado deportivo y escuelas provinciales de

formación de atletas, lo que se manifiesta negativamente en la identificación, detección, selección, promoción y retención de talentos deportivos.

También plantean que los directivos no poseen una formación específica para ejercer sus responsabilidades, interactúan con sus subordinados a partir de su experiencia personal y con los métodos que consideran más efectivos, es decir, se manifiesta carencia de armonía en los modos de actuación de los distintos niveles de dirección y gestión de la Selección Deportiva, relacionados con las formas organizativas que requiere una actividad que, en la medida en que evoluciona, se hace más compleja.

Pues es necesario transformarlas, diseñar y aplicar formas particulares o específicas de gestión que respondan a las necesidades del Deporte en la base que tengan en cuenta sus características y complejidades, que se establezca una relación funcional entre los factores externos, la tecnología y los procesos internos de la organización, ya que las existentes no tienen correspondencia con las exigencias actuales y no reportan los beneficios esperados en la provincia de Matanzas, donde las peculiaridades de la gestión de la Selección Deportiva , desde otros enfoques, no han sido suficientemente investigadas.

Si se consideran los criterios anteriormente expresados, se aprecia que las crecientes demandas para mejorar el deporte en nuestro territorio manifiestan contradicción entre las formas de la gestión que se aplican para la Selección Deportiva en la provincia, ante un entorno cambiante y la carencia de formas específicas de realizarla con eficacia y eficiencia.

1.2 Enfoques existentes

Como parte de esta investigación, en una fase inicial de la misma, se realizó una búsqueda con el objetivo de verificar la existencia de sistemas informáticos similares y que cumpliera total o parcialmente las exigencias del cliente. De acuerdo a los resultados, a continuación, se realiza una breve descripción de algunos sistemas informáticos, especializados en el proceso de gestión de atletas y pruebas físicas:

1.2.1 Sistema de gestión de atletas a partir de la arquitectura de microservicios

Es un sistema de gestión de atletas donde la información sea accesible a partir de peticiones a un microservicio, cuenta con una pequeña interfaz de usuario que coopera con el microservicio. Este sistema está pensado para la selección de los atletas de un equipo de atletismo de cara a escoger a los mejores para una competición colectiva.

Con esta aplicación, se permite tener la información actualizada y llevar un seguimiento de cada uno de los integrantes de un equipo de atletismo. La solución se ha construido a partir de software libre alcanzando una base bastante sólida sobre la que es viable, a partir de futuros desarrollos, incorporar nuevas funcionalidades que hagan de este sistema una aplicación que satisfaga todas las necesidades de un entrenador de atletismo profesional.(García Gómez, 2020)

1.2.2 Sistema de gestión y seguimiento de los entrenamientos para los deportistas de alto de rendimiento en la Provincia de Los Ríos

Esta aplicación web, permite realizar la gestión y seguimiento de los entrenamientos de mejor manera optimizando el tiempo de proceso requerido para ser revisado cada deportista, debido a que estos procesos en la actualidad se realizan de manera metódica, es decir, de manera manual. El sistema servirá como soporte para que los entrenadores puedan controlar y analizar los entrenamientos de cada uno de sus deportistas, sin la necesidad de tener que perder mucho tiempo en la captación y procesamiento de la información resultante de realizar un entrenamiento. Esta aplicación nace con la intención de ser una herramienta útil tanto para encargados del departamento técnico metodológico, entrenadores y deportistas, permitiendo la automatización de procesos operativos, suministrando la información necesaria para que se puede tomar las decisiones oportunas y acertadas; ya que la implantación permite lograr ventajas significativas en la obtención de los objetivos propuestos.(Troya Matos, 2020)

1.2.3 Diseño de un sistema de Gestión de la Calidad Total en el ámbito del deporte. Modelo MEXD de Excelencia Deportiva

Esta tesis doctoral propone la construcción de una herramienta que permita evaluar la calidad percibida de los deportistas y usuarios de instalaciones y eventos deportivos, y que suponga un medio para mejorar los estándares de calidad en las actividades y servicios deportivos ofertados. A partir de este estudio y de la investigación realizada de los modelos de gestión existentes, se propone la definición del concepto de Excelencia Deportiva y los principios fundamentales de la misma, así como la creación de un Modelo de Excelencia Deportiva (MEXD) exclusivo para todas las organizaciones del entorno deportivo.

La base del modelo es la autoevaluación. Aunque la autoevaluación suele ser aplicada al conjunto de la organización o complejo deportivo, también puede evaluarse un evento deportivo o instalación deportiva de forma aislada. La autoevaluación definida en el modelo MEXD permite a las organizaciones y sus directivos identificar claramente sus puntos fuertes y sus áreas de mejora.(Salazar Chang, 2015)

1.2.4 Sistema Informático para la Administración de Expedientes y Seguimiento de Planes de Entrenamiento del INDES

Este sistema es utilizado por el Departamento de Alto Rendimiento como apoyo en sus actividades laborales y permitirá realizar diferentes informes que faciliten la toma de decisión para las autoridades superiores de la institución. Este sistema registra la información general de los atletas de alto rendimiento que están asignados en las diferentes federaciones deportivas y permite que los metodólogos del Departamento de Alto Rendimiento registren el seguimiento que realizan a los entrenamientos de los atletas. Además registra los datos de los entrenadores, de las federaciones deportivas, de estímulos deportivos y de los diferentes eventos deportivos en los que participan los atletas.

Para el desarrollo del sistema se ha utilizado el ciclo de vida de desarrollo de proyectos en cascada, la técnica de entrevistas para el levantamiento de requerimientos y análisis de la situación actual, casos de uso para modelar el sistema informático y software libre para la

elaboración del software del sistema informático.(VALENCIA, 2013)

A pesar que se identificaron diferentes sistemas informáticos que tratan el tema de la gestión de atletas y pruebas físicas, no constituyen una solución completa al problema planteado es por ello que se decide implemntar un sistema informático que si cumpla con las necesidades y el flujo de trabajo para la selección de atletas en la provincia de Matanzas.

1.3 Metodologías para el desarrollo de software

En la actualidad existen una gran cantidad de metodologías para el desarrollo de software, separadas en dos grandes grupos; las metodologías tradicionales o pesadas y las metodologías agiles. Las metodologías tradicionales se basan en las buenas prácticas dentro de la ingeniería del software, siguiendo un marco de disciplina estricto y un riguroso proceso de aplicación, entre las metodologías tradicionales se encuentran RUP (*Rational Unified Process*) y MSF (*Microsoft Solution Framework*) siendo RUP la metodología más utilizada.(Sánchez Muguercia, 2013) Las metodologías ágiles, en cambio, representan una solución a los problemas que requieren una respuesta rápida en un ambiente flexible y con cambios constantes, haciendo caso omiso de la documentación rigurosa y los métodos formales, dentro de este grupo se encuentra la metodología *eXtreme Programming* (XP) o Programación Extrema. (Beck, Hendrickson y Fowler, 2001), (Maida y Pacienzia, 2015)
A continuación se muestra una tabla en la cual se comparan ambos grupos:

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios

Continúa en la siguiente página

Tabla 1.1 Continuación de la página anterior

Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo.
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Cierta resistencia a los cambios	Preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente por el equipo
La arquitectura del software es esencial y se expresa mediante modelos normas	Menos énfasis en la arquitectura del software

Tabla 1.1: Fuente: Tomada de (Letelier, 2006)

Se decide utilizar una metodología de desarrollo de software ágil y no una tradicional porque es más flexible y adaptable, permite cambios de ser necesarios de los requisitos del proyecto y las necesidades existentes, no enfatiza en la arquitectura de software. Teniendo en cuenta que el proceseo de desarrollo es relativamente corto, es más factible utilizar una metodología ágil.

1.3.1 Metodologías ágiles de desarrollo

En la actualidad existen un sin número de metodologías ágiles, unas más populares que otras, cada una aportando al desarrollo ágil distintos métodos que ayudan a mejorar de una manera eficaz la calidad del software. Entre las metodologías ágiles más utilizadas se encuentran Programación Extrema (XP), Scrum, Kanban y Lean.

1.3.2 Elección de la metodología de desarrollo de software

Metodología de desarrollo XP

La programación extrema es un proceso ágil de desarrollo de software, enfocada a las buenas prácticas de codificación, una clara comunicación y al trabajo en equipo. Está concebida para proyectos medianos y pequeños donde los requisitos son cambiantes. Por lo tanto, tiene una serie de reglas y recomendaciones que se pueden dividir en planeación y gestión, diseño, codificación, y pruebas para producir un software. En la planeación y gestión se utilizan historias de usuario, en vez de los casos de uso, para definir el funcionales del software.

El diseño debe ser simple para lo cual se usan tarjetas Clase-Responsabilidad-Colaborador (CRC). También se implementan metáforas que permitan explicar la estructura del sistema a los nuevos integrantes del equipo. La codificación se realiza en parejas, es estandarizada por el equipo de trabajo. Además, se realizan liberaciones frecuentes de versiones. Por último, están las pruebas funcionales, donde se evalúa si la historia de usuario fue implementada correctamente (Acceptance tests). También, se realizan las pruebas unitarias que deben ser verificadas para todo el código del proyecto. (Jiménez-Builes, Ramírez-Bedoya y Branch-Bedoya, 2019)

Se decide usar XP como metodología de desarrollo de software, una de las razones principales es su enfoque en la calidad del código. Con XP, el desarrollo de pruebas unitarias es una práctica esencial que se integra en la programación diaria, esto significa que no solo se está creando software que funcione, sino que también de que sea robusto, escalable y fácil de mantener en el futuro. Las reuniones de seguimiento, las revisiones de código y las pruebas continuas permiten mantener en sintonía los objetivos del proyecto y ajustar el trabajo en consecuencia. Permite responder rápidamente a los cambios en los requisitos que vayan surgiendo.

1.4 Tecnologías y herramientas a utilizar

Es esencial conocer las herramientas y tecnologías que se utilizarán en el proceso de desarrollo de software. No solo es importante saber cómo seleccionar las adecuadas para un proyecto en particular, sino también saber cómo utilizarlas de manera efectiva. Es importante comprender sus capacidades, limitaciones y requisitos de implementación, esto ayudará a asegurar que se elija la herramienta adecuada para el trabajo y que se pueda desarrollar de manera efectiva el software. Además, conocer las herramientas y tecnologías utilizadas en el desarrollo de software puede permite trabajar de manera más eficiente y efectiva. Al conocer las mejores prácticas y las técnicas avanzadas de la herramienta, se puede optimizar el proceso de desarrollo y mejorar la calidad del código. Es necesario escoger herramientas y tecnologías que permitan crear aplicaciones de manera más eficiente y efectiva. Estas pueden incluir lenguajes de programación, plataformas de desarrollo, frameworks, bibliotecas, herramientas de gestión de versiones, entre otras. Conocer y dominar estas herramientas y tecnologías permite crear software más rápido, con menos errores y de mayor calidad, lo que a su vez puede mejorar la experiencia del usuario y aumentar la eficiencia de los procesos.

1.4.1 Lenguaje de Modelo

Es un lenguaje artificial usado para expresar información, conocimiento, o sistemas, en una estructura definida por un conjunto de reglas consistentes. Desde el punto de vista de la ingeniería de software proporciona una forma estándar de describir el diseño y la funcionalidad de un sistema (Parra, 2018) .

1.4.1.1 *Elección del lenguaje de modelado*

Se selecciona como lenguaje el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*), debido a que se emplea para visualizar, especificar, construir y documentar los artefactos de la solución propuesta.

La formalización de los diagramas del UML permite que cada modelo de sistemas se refine, admitiendo la inclusión y la refinación de las relaciones entre los elementos, chequeando

la consistencia interna de cada uno de los elementos, y verificando la interconexión entre los elementos. UML surge como una herramienta de gran aceptación cuando es necesario soportar el diseño y la implementación de una solución automatizada, que subyace en un modelo de gestión de cualquier sistema. Para ello se debe tener la documentación apropiada para su desarrollo y su mantenimiento subsiguiente o eventuales modificaciones. Lo anterior resulta deseable y debe tenerse en cuenta en las representaciones visuales del sistema para su adecuada operación y un mejor entendimiento de los diseños(Silva et al., 2018)

1.4.2 Herramienta CASE

CASE es el acrónimo de (*Computer Aided Software Engineering*), ó ingeniería de software asistida por computadora, y se refiere al uso de programas de cómputo para organizar y administrar el desarrollo de software, sobre todo en aquellos proyectos donde se involucre una cantidad considerable de recursos y personal, tarea que para el gestor de proyectos representa una inversión considerable de tiempo, por tal motivo estas herramientas tienen como propósito el fungir como apoyo durante el ciclo de vida de todo el proyecto incluyendo todos sus componentes.(Vázquez et al., 2018)

1.4.2.1 *Visual Paradigm*

Visual Paradigm es una herramienta de diseño y gestión de sistemas informáticos potente, multiplataforma y fácil de usar. Visual Paradigm proporciona a los desarrolladores de software una plataforma de desarrollo de vanguardia para construir aplicaciones de calidad más rápido, mejor y más barato. Facilita una excelente interoperabilidad con otras herramientas CASE y con la mayoría de los principales IDEs, lo que hace que todo el proceso de desarrollo Modelo-Código-Despliegue sea una solución única (Paradigm, 2022).

Soporta multitud de estándares de modelado como UM, SysML, ERD, DFD, BPMN. Es una herramienta que facilita y agiliza el desarrollo de los proyectos basados en experiencias de usuario apoyándose en la identificación de los casos, requisitos o flujos de acontecimientos (Oscar, 2013).

1.4.2.2 *Elección de la herramienta CASE*

Se seleccionó como herramienta CASE el *Visual Paradigm* porque soporta el ciclo de vida completo del desarrollo de software, permite modelar los diagramas de la solución propuesta, realizar ingeniería tanto directa como inversa.

Muestra una interfaz de usuario intuitiva, amplia gama de características, integración con otras herramientas, capacidad de colaboración en equipo. La herramienta es fácil de usar y ofrece una amplia variedad de opciones de modelado, incluyendo diagramas UML, diagramas de flujo de datos y diagramas de entidad-relación, se integra bien con otras herramientas y tecnologías, lo que facilita la colaboración en equipo y la comunicación..

1.4.3 Marco de Trabajo

En el desarrollo de software, un framework es una composición conceptual y tecnológica con un soporte bien definido, habitualmente con módulos de software concretos, en base a la cual otro proyecto de software puede ser fácilmente organizado y desarrollo. Para el desarrollo de software se hace imprescindible el uso de frameworks ya que incluyen bibliotecas, lenguaje, soportes entre otras herramientas la cuales facilitan el desarrollo de aplicaciones web(Ríos et al., 2016). A continuación, se describe de manera breve algunas de las características de estos frameworks:

1.4.3.1 Django

Pyramid, y Web2PY Django posee un gran número de librerías para manejar tareas comunes del desarrollo web tales como la autenticación de usuarios, administración de contenidos, los mapas del sitio, los feed RSS y mucho más. Otro aspecto destacable es la seguridad que presenta, Django se encuentra preparado para evitar muchos errores de seguridad comunes como la inyección de SQL, el cross-site scripting, la falsificación de peticiones entre sitios y el clickjacking, además, sistema de autenticación de usuarios proporciona una forma segura de gestionar las cuentas y las contraseñas de los usuarios. Por otra parte, posee la capacidad de soportar demandas de tráfico muy intensas, volviéndose uno de los frameworks más escalables de la actualidad. El patrón de arquitectura que emplea es

Model-View-Template que sigue el mismo principio del Modelo-Vista-Controlador, con la diferencia que la capa de presentación son las plantillas (Template) y la capa donde se encuentra toda la lógica de programación se denomina vista (View) (Espinosa-Hurtado, 2021).

Django es un marco de trabajo web de alto nivel en Python que fomenta un desarrollo rápido y un diseño limpio y pragmático. Es gratuito y de código abierto. Este framework se caracteriza porque permite construir aplicaciones de forma rápida, segura y con menos código (en comparación con otros frameworks). Además, cuenta con una comunidad de colaboradores muy grande a nivel mundial que se encarga de realizar las debidas actualizaciones tanto del framework como de su documentación de forma diaria (Gómez García, 2018).

1.4.3.2 Elección del framework de desarrollo

Despúes de realizar una amplia evalución de distintos frameworks para el desarrollo de software, se decide utilizar Django como marco de trabajo. Es un framework web de alto nivel y de código abierto que utiliza el lenguaje de programación Python, ya familiarizado con el lenguaje y su sintaxis, hace que la curva de aprendizaje para Django sea más fácil de manejar. Se enfoca en la seguridad y la escalabilidad. Su arquitectura basada en el patrón de diseño Modelo-Vista-Template (MVT) permite una separación clara de la lógica de la aplicación. Cuenta una gran cantidad de características integradas, como la administración de la base de datos, la autenticación de usuarios y la generación de formularios, lo que facilita el desarrollo del software. Otro aspecto que facilitó la eleccion de Django es su comunidad activa y su ecosistema de herramientas y recursos, existe una gran cantidad de documentación y tutoriales disponibles en línea, además de una gran cantidad de paquetes y complementos de terceros que pueden integrarse fácilmente en una aplicación de Django. Su compatibilidad con una amplia variedad de bases de datos, servidores web y sistemas de control de versiones, hace fácil de integrar en cualquier entorno de desarrollo.

1.4.4 Entornos de desarrollo

Un entorno de desarrollo integrado, conocido también como IDE (*Integrated Development Environment*) por sus siglas en inglés, es un programa informático compuesto por un conjunto de herramientas de programación que facilitan el desarrollo de aplicaciones. Un IDE puede denominarse como un entorno de programación, esto significa que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación (Muñoz, 2012), (Jiménez, 2016).

1.4.4.1 Visual Studio Code

El IDE de Visual Studio es una plataforma de lanzamiento creativa que se puede utilizar para editar, depurar y construir código, y luego publicar una aplicación. Además del editor y el depurador estándar que proporcionan la mayoría de los IDE, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y muchas más funciones para mejorar el proceso de desarrollo de software. Visual Studio tiene soporte para el entorno Windows y macOS, ambas con las mismas funciones, y está optimizado para el desarrollo de aplicaciones móviles y multiplataforma. Existen tres ediciones disponibles: Community (libre), Professional y Enterprise, estas dos últimas son de pago.(Johnson, 2012)

1.4.4.2 Elección del entorno de desarrollo

Visual Studio es el IDE más completo para la plataforma .NET y C++ y posee extensiones para diferentes lenguajes de programación. Uno de los principales inconvenientes es que la versión gratuita que ofrece Microsoft está limitada en cuanto a funciones y extensiones, además, ocupa un gran espacio en el disco duro tanto el instalador como el producto instalado y por la gran cantidad de servicios que ejecuta luego de instalarse requiere una computadora con elevadas prestaciones. Por otro lado el IDE PyCharm está enfocado principalmente para el lenguaje de programación Python; el principal inconveniente es que la versión gratuita que se ofrece por el desarrollador está limitada solamente a desarrollar aplicaciones de consola con el lenguaje Python, con lo cual queda descartado su uso para

llevar a cabo la presente investigación. Finalmente el IDE Visual Studio Code reúne muchas características que lo vuelven el IDE ideal para muchos desarrolladores: gratuito en su totalidad, presenta una gran cantidad de extensiones para casi cualquier lenguaje existente, extremadamente ligero y en dependencia del tipo de software a desarrollar se seleccionan las extensiones necesarias, con lo cual el espacio en disco no resulta ningún problema.

Por todos los elementos anteriores se selecciona Visual Studio Code con las extensiones django-intellisense, Django, Python, Jupyter y HTML y CSS Support como entorno de desarrollo integrado para ser utilizado en la elaboración de la solución.

1.4.5 Lenguajes de programación

Un lenguaje de programación es un idioma diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Definido además como un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una computadora. (**LOUDEN**; **KENNETH. Lenguajes de programaciÃşn: Principios y prÃąctica. s.l: Cengage 2004**)

Los lenguajes de programación son la herramienta básica de construcción de programas, como lo son el machete y el azadón para un campesino, el pico y la pala para un constructor. (**challenger2014lengua**)

1.4.5.1 Python

Python cuenta con facilidades para la programación orientada a objetos,imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas. Fue basado en el lenguaje ABC y se dice que fue influenciado por otros como C, Algol 60, Modula-3 e Icon, Es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible. Python ha ido ganando en adeptos en comunidades como la de software libre, científica y educacional, por su sencillez y posibilidad de

concentrarse en los problemas actuales. (**challenger2014lengua**).

Es el lenguaje seleccionado para el desarrollo del sistema debido a que es el lenguaje que integra el framework web Django.

1.4.5.2 HTML

HTML (HyperText Markup Language) es un lenguaje de marcado utilizado para crear y diseñar páginas web. Se trata de un lenguaje de programación básico que se utiliza para estructurar y presentar el contenido de una página web, incluyendo texto, imágenes, enlaces y otros elementos multimedia. HTML se basa en etiquetas de marcado que se utilizan para definir la estructura de una página web, así como para indicar el formato y estilo de los elementos de la página. HTML es un estándar web que es interpretado por los navegadores web para mostrar la página web al usuario final. (Vértice, 2009)

1.4.5.3 CSS

Hojas de Estilo en Cascada (Cascading Style Sheets) o CSS, es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios lenguajes basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.

La versión actual de CSS es CSS3, le preceden CSS1 y CSS2.1. Desde CSS3, el alcance de las especificaciones se incrementó de forma significativa y el progreso de los diferentes módulos de CSS comenzó a mostrar varias diferencias, lo que hizo más efectivo desarrollar y publicar recomendaciones separadas por módulos. En vez de versionar las especificaciones de CSS, la W3C actualmente realiza una captura de las ultimas especificaciones estables de CSS.

CSS es utilizado para diseñar y dar estilo a las páginas web, por ejemplo, alterando la fuente, color, tamaño y espaciado del contenido, dividirlo en múltiples columnas o agregar ani-

maciones y otras características decorativas. Este módulo proporciona un suave comienzo hacia el dominio de CSS con los conceptos básicos acerca de su funcionamiento, la sintaxis y la manera en que puedes comenzar a utilizarlo para agregar estilos al HTML (Foundation, 2022).

1.4.5.4 JavaScript

JavaScript no es más que un sencillo lenguaje de programación, que presenta una característica especial: sus programas, llamados comúnmente scripts, se en las páginas HTML y se ejecutan en el navegador (Netscape Navigator y Microsft Explorer). Estos scripts normalmente consisten en unas funciones que son llamadas desde el propio HTML cuando algún evento sucede. De ese modo, podemos añadir efectos como que un botón cambie de forma al pasar el ratón por encima, o abrir una ventana nueva al pulsar en un enlace. (Navarrete, 2006)

1.4.5.5 jQuery

jQuery es una biblioteca de JavaScript diseñada para simplificar la manipulación y el manejo de documentos HTML, el manejo de eventos, las animaciones y las interacciones con el servidor en el lado del cliente.

jQuery permite a los desarrolladores escribir código más conciso y fácil de leer que el JavaScript estándar, lo que a su vez facilita el mantenimiento y la escalabilidad del código. Además, jQuery es compatible con la mayoría de los navegadores web modernos y es fácil de integrar en los proyectos existentes. Con jQuery, se puede crear efectos visuales, manipular el contenido de la página web, interactuar con el usuario y hacer solicitudes al servidor sin tener que escribir grandes cantidades de código JavaScript. (Alvarez, 2010)

1.4.6 Servidor Web

Un servidor web es un programa de software que se ejecuta en una computadora y que se utiliza para entregar contenido web a través de Internet. El servidor web recibe solicitudes de los clientes (navegadores web) y responde a esas solicitudes enviando el contenido

solicitado al cliente. El servidor web está diseñado específicamente para manejar solicitudes HTTP (Hypertext Transfer Protocol), que es el protocolo utilizado para transferir datos entre el servidor y el cliente. Los servidores web pueden entregar diferentes tipos de contenido, como páginas web estáticas (HTML, CSS, imágenes), archivos multimedia (audio, video) y aplicaciones web dinámicas (JavaScript, PHP, Python). Para entregar estos contenidos, el servidor web utiliza un conjunto de reglas y configuraciones específicas que le permiten saber cómo procesar las solicitudes y cómo entregar el contenido correspondiente.(Andreu, 2011)

1.4.7 Servidor Web Apache

Apache es el servidor Web con mayor presencia en el mercado mundial. Aunque su configuración es relativamente sencilla, fortalecer sus condiciones de seguridad implica entender y aplicar un conjunto de reglas generales conocidas, aceptadas y disponibles. Se ejecuta en sistemas operativos como Linux, Windows, MacOS y otros. Es compatible con una amplia variedad de lenguajes de programación, como PHP, Python, Perl y Ruby, y puede manejar diferentes tipos de contenidos, desde páginas estáticas hasta aplicaciones web dinámicas. Apache también es altamente personalizable y tiene una gran cantidad de módulos y complementos disponibles que pueden agregar funcionalidad adicional al servidor web. Estos módulos incluyen herramientas para la compresión de contenido, la optimización de la velocidad de carga de las páginas web y la gestión de la caché. (Montoya, Uribe y Rodríguez, 2013)

1.4.8 Elección del servidor web

Se decide usar Apache como servidor web, debido a su compatibilidad con Django, puede configurar fácilmente el módulo mod wsgi para alojar la aplicación web de Django en Apache sin tener que realizarse cambios importantes. Presenta un sólido conjunto de características de seguridad integradas, como la autenticación de usuarios, la gestión de permisos y la protección contra ataques de denegación de servicio permitiendo que la aplicación web cumpla con los patrones de segurudad, tiene la capacidad para manejar grandes volúmenes de tráfico web y es altamente escalable, lo que significa que se puede ajustar la

configuración según las necesidades de la aplicación web y manejar grandes volúmenes de solicitudes de manera efectiva.

1.4.9 Sistema Gestor de Base de Datos

Un sistema gestor de base de datos (SGBD) es un software que permite la gestión y organización de grandes cantidades de datos. Los SGBD están diseñados para administrar la creación, almacenamiento, modificación y recuperación de datos en un formato estructurado que pueda ser utilizado por aplicaciones y usuarios. En general, un SGBD permite la gestión eficiente y segura de grandes cantidades de datos, lo que es esencial para empresas y organizaciones que necesitan almacenar y procesar información crítica. (Vega et al., 2019)

1.4.9.1 MySQL

MySQL es un popular sistema de gestión de bases de datos SQL de código abierto desarrollado, distribuido y respaldado por Oracle Corporation. MySQL gestiona una colección estructurada de datos. Una base de datos MySQL le ayuda a agregar, acceder y procesar los datos almacenados en la base de datos. MySQL almacena datos en tablas separadas. Las estructuras de la base de datos están organizadas en archivos físicos optimizados para la velocidad. El modelo lógico, con objetos como bases de datos, tablas, vistas, filas y columnas, ofrece un entorno de programación flexible. La parte SQL de "MySQL"significa "Lenguaje de consulta estructurado", que es el lenguaje estandarizado más común utilizado para acceder a las bases de datos. (Christudas y Christudas, 2019)

1.4.9.2 Eleccióonn del sistema gestor de base de datos

Se decide usar MySQl como gestor de base de datos debido a su compatibilidad con Django, el cual es compatible con varios sistemas gestores de bases de datos, pero MariaDB es una opción muy eficaz debido a su facilidad de uso y rendimiento confiable, no será necesario aprender un nuevo conjunto de comandos y sintaxis, destacar que ofrece algunas mejoras y características adicionales, como una mayor escalabilidad y mejor rendimiento en entornos de alta carga. Proporcionando la incorporación de nuevos motores de búsqueda

para permitir mayor escalabilidad y mejores velocidades al momento de realizar consultas a la base de datos.

1.4.10 Sistema de Control de Versiones

Los sistemas de control de versiones son aplicaciones que ayudan al proceso de desarrollo de software, facilitando la gestión del control de versiones de los archivos de código fuente generados por los desarrolladores, proporcionando herramientas para la fusión y generación de una nueva versión de un proyecto, permitiendo que múltiples desarrolladores trabajen en el mismo proyecto sin ocasionar pérdida de datos o bloqueos de archivos. Además, permiten recuperar archivos generados previamente, los cuales pueden ser utilizados para solucionar errores del sistema. (Leal, Sosa y Leal, 2012)

1.4.11 Git

Git permite llevar en paralelo varias versiones del mismo software, utilizando un modelo distribuido en el que cada desarrollador tiene una copia completa del repositorio y puede trabajar de forma independiente sin tener que estar conectado a un servidor central. Además, incluye herramientas integradas para la gestión de ramas y etiquetas, la fusión de cambios y la resolución de conflictos. Es compatible con múltiples sistemas operativos, incluyendo Windows, Mac OS X y Linux, lo que lo hace una opción versátil para proyectos de software multiplataforma. (Lopez-Pellicer et al., 2015)

1.4.12 Elección del Sistema de Control de Versiones

Se utiliza Git como Sistema de Control de Versiones dado que existe una gran cantidad de recursos disponibles en línea, incluyendo documentación, tutoriales y comunidades de apoyo que pueden ayudarte a aprender y utilizar Git de manera efectiva, permite trabajar de forma independiente sin tener que estar conectado a un servidor central. Teniendo en cuenta su rápidez y eficiencia permite que el control del desarrollo de software sea de forma segura y fácil.

1.5 Conclusiones parciales del capítulo

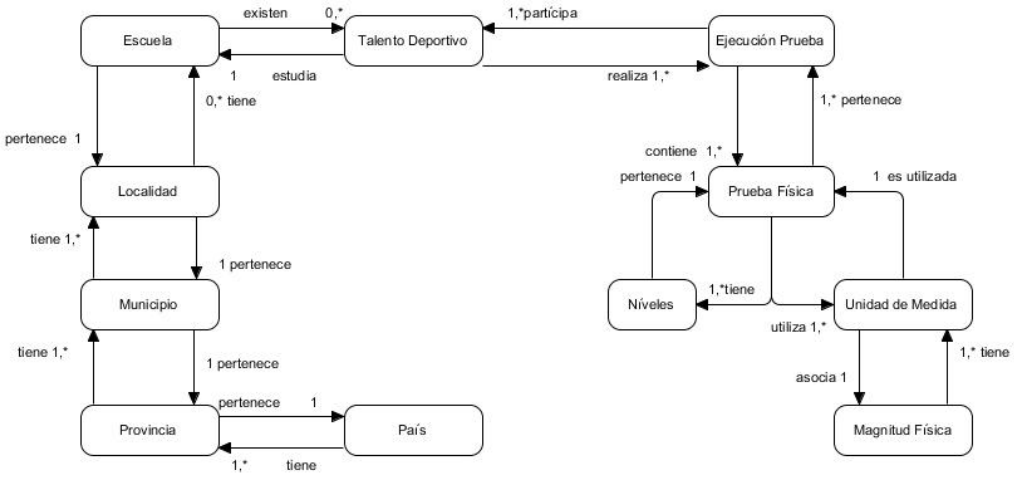
Después de analizar los conceptos fundamentales asociados a la selección de atletas en la provincia de Matanzas, se puede concluir que la gestión de pruebas físicas es un proceso crucial en la identificación y selección de talentos deportivos. Se ha demostrado que las aplicaciones de gestión de pruebas físicas pueden mejorar significativamente la eficacia y eficiencia del proceso de selección de atletas. Se hizo un profundo analisis en softwares similares que den solución a la problemática, sin obtener resultados que cumpla la totalidad de los objetivos Se ha seleccionado una serie de tecnologías, metodologías y softwares para el desarrollo de la solución propuesta, que se consideran las más adecuadas para satisfacer las necesidades de la provincia de Matanzas en términos de gestión de pruebas físicas para la selección de atletas.

Capítulo 2: Análisis y diseño del sistema

Este capítulo describe el proceso de análisis y diseño de la solución propuesta para abordar el problema identificado. La metodología de desarrollo elegida ha sido el marco guía para llevar a cabo este proceso. En primer lugar, se presenta el modelo de dominio, el cual describe las entidades involucradas en el proceso de negocio y facilita la comprensión de los conceptos clave relacionados. A continuación, se exponen los artefactos más importantes que describen el flujo normal de eventos en el sistema, seguido por la descripción de la solución propuesta, que incluye los requisitos funcionales y no funcionales identificados. Finalmente, se define la arquitectura que se utilizará para implementar la solución propuesta. Este capítulo proporciona una visión detallada del proceso de análisis y diseño, y sienta las bases para la implementación de la solución propuesta en el siguiente capítulo.

2.1 Modelo de dominio

Un modelo de dominio muestra las clases conceptuales significativas en un dominio del problema. Su utilidad radica en ser una forma de "inspiración" para el diseño de los objetos software. Es entrada para muchos de los artefactos que se construyen en un proceso software. Se centra en las abstracciones relevantes, vocabulario del dominio e información del dominio, es el artefacto clave del análisis orientado a objetos. (Vázquez-Ingelmo y García-Peñalvo, 2019)



2.2 Propuesta del sistema

Se propone un sistema llamado Sistema para la gestión de pruebas físicas del proceso de selección deportiva en la provincia de Matanzas. Para su desarrollo se utilizaron herramientas libres cuyas funcionalidades permitan la gestión de dichas pruebas con el fin de captar adecuadamente los mejores atletas.

2.3 Fase de exploración y planificación

La fase de exploración es una de las primeras etapas en el proceso de desarrollo de software. En esta fase, el equipo de desarrollo se enfoca en comprender en profundidad el problema que se desea resolver e identificar las necesidades y expectativas de los usuarios del sistema. El equipo de desarrollo de XP trabaja en estrecha colaboración con los usuarios para comprender en profundidad el problema que se desea resolver. Se llevan a cabo actividades como entrevistas, observación de usuarios en su entorno de trabajo, y la creación de historias de usuario que describen las necesidades de los usuarios en un lenguaje claro y conciso. Esta etapa también incluye la definición de los objetivos del proyecto y la identificación de los aspectos técnicos y arquitectónicos que deberán abordarse en el desarrollo. (Aranguren Peraza, 2007)

2.3.1 Requisitos funcionales

Requisitos funcionales se refieren a las funcionalidades o capacidades específicas que se espera que el software proporcione a los usuarios o al sistema. Estos requisitos describen las acciones que el software debe ser capaz de realizar, y establecen los criterios para evaluar si el mismo cumple con las expectativas y necesidades de los usuarios, derivan de la comprensión del problema y las necesidades de los usuarios, y se definen en función de los objetivos del proyecto. Estos requisitos pueden incluir acciones específicas que el software debe ser capaz de realizar, como la capacidad de crear, modificar y eliminar registros de datos, la capacidad de generar informes y estadísticas, o la capacidad de interactuar con otros sistemas. (Toro y Jiménez, 2000)

1. El Sistema debe permitir gestionar usuario
    - Añadir usuario
    - Eliminar usuario
    - Actualizar usuario
    - Listar usuario
2. El sistema debe permitir autenticar usuario
    - Iniciar sesión
    - Cambiar clave de acceso
    - Cerrar sesión
3. El Sistema debe permitir gestionar rol
    - Añadir rol
    - Eliminar rol
    - Actualizar rol
    - Listar rol
4. El Sistema debe permitir gestionar sexo

- Añadir sexo
- Eliminar sexo
- Actualizar sexo
- Listar sexo

5. El Sistema debe permitir gestionar atleta
   - Añadir atleta
   - Eliminar atleta
   - Actualizar atleta
   - Listar atleta
6. El Sistema debe permitir gestionar unidad de medida
   - Añadir unidad de medida
   - Eliminar unidad de medida
   - Actualizar unidad de medida
   - Listar unidad de medida
7. El Sistema debe permitir gestionar magnitud física
   - Añadir magnitud física
   - Eliminar magnitud física
   - Actualizar magnitud física
   - Listar magnitud física
8. El Sistema debe permitir gestionar prueba
   - Añadir prueba
   - Eliminar prueba
   - Actualizar prueba

- Listar prueba

9. El Sistema debe permitir gestionar país
    - Añadir país
    - Eliminar país
    - Actualizar país
    - Listar país
10. El Sistema debe permitir gestionar provincia
    - Añadir provincia
    - Eliminar provincia
    - Actualizar provincia
    - Listar provincia
11. El Sistema debe permitir gestionar municipio
    - Añadir municipio
    - Eliminar municipio
    - Actualizar municipio
    - Listar municipio
12. El Sistema debe permitir gestionar localidad
    - Añadir localidad
    - Eliminar localidad
    - Actualizar localidad
    - Listar localidad
13. El Sistema debe permitir gestionar escuela
    - Añadir escuela

- Eliminar escuela
- Actualizar escuela
- Listar escuela

14. El Sistema debe permitir gestionar reporte

    - El sistema debe permitir generar un reporte con todas las pruebas realizadas a un determinado atleta
    - El sistema debe permitir generar un reporte con todos los atletas que han realizado una determinada prueba
    - El sistema debe permitir exportar reportes a archivos de formato PDF

15. El sistema debe permitir generar gráficas

2.3.2 Requisitos no funcionales

Los Requisitos No Funcionales (RNF) de software forman una parte significativa de la especificación de requisitos y en algunos casos estos son críticos para el éxito del producto. Con frecuencia estos requisitos son ignorados o subestimados debido a que para muchos proyectos estos implican una cantidad considerable de trabajo y esfuerzo; resultan ser más complejos y requieren un mayor nivel de conocimiento. No se centran únicamente en qué debe hacer el sistema, sino en cómo debe hacerlo. Estos requisitos se refieren a características y criterios que no están directamente relacionados con la funcionalidad del sistema, pero que son igualmente relevantes para su éxito y efectividad. Los requisitos no funcionales ayudan a garantizar que el sistema sea confiable, seguro, eficiente y fácil de usar, lo cual es esencial para la satisfacción de los usuarios y el éxito general del proyecto. (Molina Hernández, Granda Dihigo y Velázquez Cintra, 2019)

1. **RNF-1 Interfaz:**

    a) Clara y concisa. No debe dar lugar a la confusión del usuario y debe seguir los estándares de diseño de interfaces de Google.

b) Las interfaces visuales del sistema deben ser diseñadas para ser visualizadas en modo *landscape*.

c) La interfaz de usuario del sistema deberá ser diseñada de forma tal que permita el aprovechamiento del espacio.

d) En la pantalla principal se tendrá acceso a la mayoría de los módulos con que contará el sistema de manera sencilla y con un solo clic.

e) Los componentes de la interfaz de usuario para la entrada de datos, de ser posible, deberán ser configurados de manera que el riesgo de entrada de datos inválidos por parte del usuario disminuya.

2. **RNF-2 Estabilidad:** El sistema debe ser capaz de manejar los errores ocurridos durante la ejecución de la misma y avisando de la naturalidad del error.

3. **RNF-3 Rendimiento:** El sistema debe desempeñar su función de una manera fluida. Se debe buscar la experiencia de uso más agradable para el usuario.

4. **RNF-4: Usabilidad**

   a) El sistema deberá visualizarse correctamente en cualquier dispositivo (monitor o pantalla de lap)cuyas dimensiones no sean menoreas a.....

   b) La interfaz visual del sistema debe ser atractiva y sencilla, permitiendo al usuario facilidad de uso y entrenamiento.

   c) El sistema deberá ser capaz de brindar información al usuario mediante mensajes, para ayudarlo y guiarlo durante su interacción con la aplicación.

5. **RNF-6: Ayuda y documentación:** Se brindarán manuales de ayuda que documenten cómo trabajar de forma adecuada con el sistema.

6. **RNF-7: Hardware:**Servidor y cliente

7. **RNF-8: Software:**Navegador**

2.4 Historias de Usuarios

A continuación se muestra la descripción de cada una de las Historias de Usuarios definidos en el Sistema

HISTORIA DE USO			
Orden	HU_1.01	**Nombre**	Iniciar Sesión
Riesgo	Bajo	**Prioridad**	Alta
Iteración	1	**Puntos estimados**	0.4
Descripción	El sistema debe brindar la funcionalidad de iniciar la sesión para cada usuario existente en el sistema. El sistema debe representar dicha funcionalidad al usuario mediante una ventana, la misma debe tener los datos que se necesiten del mismo para acceder en el sistema (nombre de usuario y clave). Dicha ventana debe de aparecer una vez que el usuario intente acceder al sistema a través del navegador web.		
Observación	El sistema no puede permitir que el usuario inicie sesión dos veces consecutivas. Para que el usuario pueda iniciar sesión nuevamente, debe previamente cerrar sesión (Ver HU_01.3). Una vez ejecutada con éxito la funcionalidad, esta deja de estar disponible en el sistema hasta que no se ejecute con éxito la funcionalidad de cerrar sesión (Ver HU_01.3).		

2.5 Estimación de esfuerzo por Historias de Usuario y plan de iteraciones

Las estimaciones de esfuerzo asociado a la implementación de las Historias de Usuario se realizan con el objetivo de lograr una planificación real en el desarrollo del sistema de gestión económica y llevar un registro de la velocidad de desarrollo, basándose principalmente en la suma de puntos correspondientes a las Historias de Usuario. La planificación se puede realizar basándose en el tiempo. La velocidad de desarrollo es utilizada para establecer cuántas Historias de Usuario se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de éstas. Se establece como medida el pun-

to estimado. Un punto estimado equivale a una semana ideal de programación. Para lograr una mejor organización del trabajo y proporcionar un desarrollo iterativo e incremental, se crea el plan de iteraciones donde se planifica el orden de desarrollo de las Historias de Usuario. Se definió realizar 9 iteraciones, su orden está determinada según las prioridades de las Historias de Usuario y las dependencias existentes entre ellas. La duración total de cada iteración dependerá de los puntos estimados de las Historias de Usuario que en él se desarrollan. Teniendo en cuenta lo expuesto anteriormente la estimación de esfuerzo de las Historias de Usuario y el plan de iteraciones queda como se muestra en la siguiente tabla:

Tabla 2.2: Estimación de esfuerzo por Historia de Usuario

Iteración	**Historia de Usuario**	**Estimación de esfuerzo**
1	Iniciar sesión Cambiar clave de acceso Cerrar sesión Listar usuarios existentes en el sistema Adicionar un usuario en el sistema Actualizar datos de un usuario en el sistema Eliminar usuario en el sistema	3 semanas
2	Listar roles existentes Adicionar rol Actualizar rol Eliminar rol Asignar rol a un usuario en el sistema Remover rol a un usuario en el sistema	2 semana
3	Listar sexos existentes Adicionar sexo Actualizar sexo Eliminar sexo	1 semana
Continúa en la siguiente página		

Tabla 2.2 Continuación de la página anterior

4	Listar países existentes Adicionar país Actualizar país Eliminar país Listar provincias existentes Adicionar provincia Actualizar provincia Eliminar provincia	3 semana
5	Listar localidades existentes. Adicionar localidad. Actualizar localidad. Eliminar localidad Listar escuelas existentes. Adicionar escuela. Actualizar escuela. Eliminar escuela. Listar modalidadades de turismo existentes.	3 semanas
6	Listar atletas existentes. Adicionar atleta. Actualizar atleta. Eliminar atleta. Listar pruebas existentes. Adicionar prueba. Actualizar prueba. Eliminar prueba.	3 semanas
Continúa en la siguiente página		

Tabla 2.2 Continuación de la página anterior

7	Listar magnitudes físicas existentes. Adicionar magnitud física. Actualizar magnitud física. Eliminar magnitud física.	2 semanas
8	Listar unidades de medidas existentes. Adicionar unidad de medida. Actualizar unidad de medida. Eliminar unidad de medida.	2 semanas
9	El sistema debe permitir generar un reporte con todas las pruebas realizadas a un determinado atleta. El sistema debe permitir generar un reporte con todos los atletas que han realizado una determinada prueba. El sistema debe permitir exportar reportes a archivos de formato PDF.	3 semanas
10	El sistema debe permitir generar gráficas.	2 semanas
Total: 10	**Total:** 55	**Total:** 25 semanas

2.6 Plan de entrega

El plan de entrega es un documento que especifica con exactitud qué Historias de Usuario serán implementadas en cada entrega del sistema y sus prioridades, de modo que también permita conocer con claridad qué Historias de Usuario serán implementadas en la próxima iteración. Debe ser negociado y elaborado en forma conjunta entre el cliente y el equipo de desarrollado durante las reuniones de planificación de entregas, la idea es hacer entregas frecuentes para obtener una mayor retroalimentación.

A continuación se muestra en el plan de entrega definido para el ciclo de desarrollo.

Tabla 2.3: Plan de entregas

Iteración	Historia de Usuario	Fecha de entrega
1	7	16de julio del 2023
2	8	6 de agosto 23
3	4	20 de agosto del 2023
4	8	27 de agosto del 2023
5	8	10 de septiembre del 2023
6	8	1 de octubre del 2023
7	4	22 de octubre del 2023
8	2	5 de noviembre del 2023
9	1	19 de noviembre del 2023
10	1	3 de diciembre del 2023

2.7 Estimación del costo

Entre los aspectos que no se puede dejar abordar dentro de este capítulo de análisis esta el del análisis económico de la solución propuesta. A continuación se realizará un desglose del coste de los elementos necesarios en esta investigación. Dichos elementos incluyen costes de personal, de hardware y de software. La investigación se realizará entre el 16 de julio del 2023 al 3 de diciembre de 2023, por lo tanto, han sido 25 semanas de trabajo. Teniendo en cuenta una jornada laboral de 8 horas tendremos un total de 1000 horas de trabajo, distribuidas entre diferentes tareas y diferentes roles profesionales que las llevan a cabo.

2.7.1 Costo de personal

La metodología de software escogida propone un equipo de desarrollo pequeño donde cada integrante tiene su rol y funciones bien definidas. Para determinar el coste del personal involucrado se va desglosar el equipo de acuerdo a la categoría de cada uno así como en la fase donde participa quedando el desglose del coste como se aprecia en la siguiente tabla:

Tabla 2.4: Coste de personal

Fase	Categoría	Horas	Coste/hora	Coste
Documentación	Analista	125	150.00 CUP	18750.00 CUP
Análisis	Analista	120	150.00 CUP	18000.00 CUP
Diseño	Diseñador	95	125.00 CUP	11875.00 CUP
Diseño gráfico	Diseñador gráfico	105	125.00 CUP	13125.00 CUP
Implementación	Programador	475	100.00 CUP	47500.00 CUP
Pruebas	Programador	80	100.00 CUP	8000.00 CUP
Total		**1000**		**109250.00 CUP**

2.7.2 Costo de hadware

Para el hardware calcularemos el coste según el período de amortización teniendo en cuenta una duración del proyecto de 25 semanas. El equipo esta formado por ordenadores y computadoras portátiles, necesitando uno de cada uno de estos dispositivos para el desarrollo.

Tabla 2.5: Coste de hardware

Equipo	Coste	Coste de amortizado
CPU	26000.00 CUP	1918.80 CUP
Monitor	4000.00 CUP	252.00 CUP
Teclado	800.00 CUP	276.96 CUP
Mouse	600.00 CUP	132.72 CUP
Computadora portátil	52000.00 CUP	7664.80 CUP
Total	83400.00 CUP	10245.28 CUP

2.7.3 Costo de software

En la realización de esta investigación se ha optado por utilizar software libre por lo que no tenemos ningún coste asociado al software.

2.7.4 Costo de total

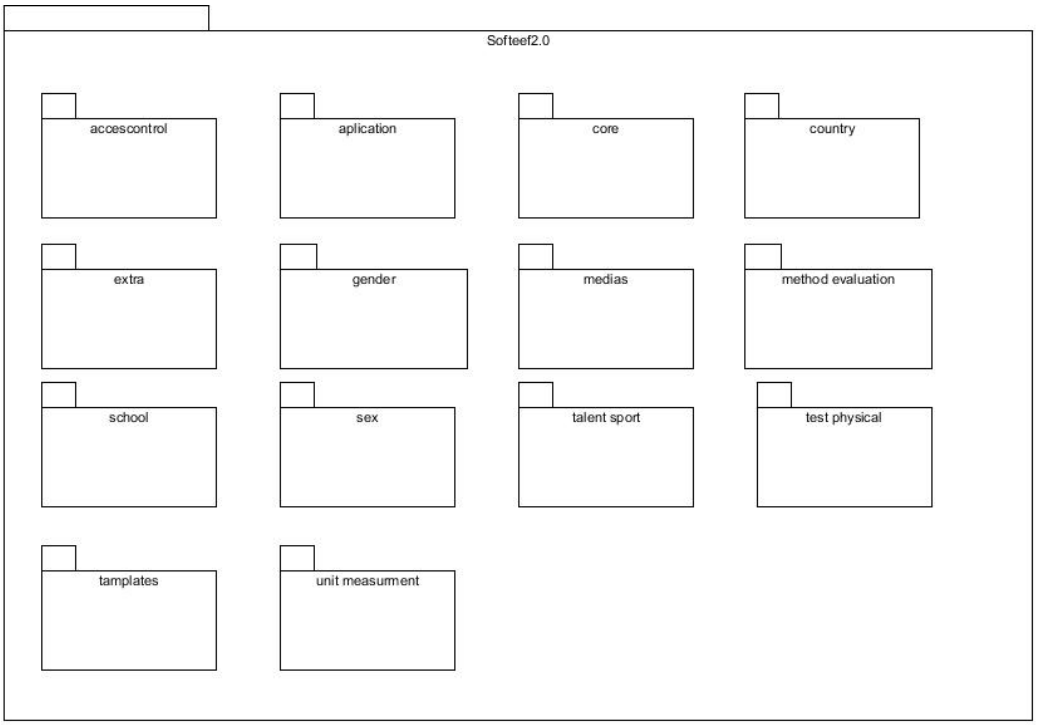
Tabla 2.6: Coste total

Tipo de coste	Total
Coste de personal	109250.00 CUP
Coste de hardware	10245.28 CUP
Coste de software	0 CUP
Total	119495sx CUP

2.8 Diagrama de paquetes

Un paquete es un grupo de elementos de un modelo. Son muy útiles en la gestión de modelos. También muy útiles en temas relacionados con la agrupación, tales como casos de uso, con el fin de facilitar la ruptura entre el trabajo en sub-equipos. Un paquete puede contener uno o más tipos de elementos del modelo, cada uno de los cuales debe tener un nombre único dentro del paquete. Un pa-quete puede ser de clases. La única regla es que cada uno de los elementos de un modelo solo puede pertenecer a un único paquete (en términos UML, un modelo es básicamente un paquete que contiene otros paquetes) (Cruz Vélez, Loayza, Huamán et al., 2005)

Cuando se crea un proyecto web con Django se genera automáticamente la estructura de carpetas necesaria para poder generar posteriormente la aplicación. Esta estructura es común para cualquier proyecto, independientemente de su tamaño y complejidad. Toda esta estructura es contenida dentro de la carpeta del proyecto que en este caso es softeef2.0. En la siguiente imagen se muestra la estructura correspondiente a la solución:

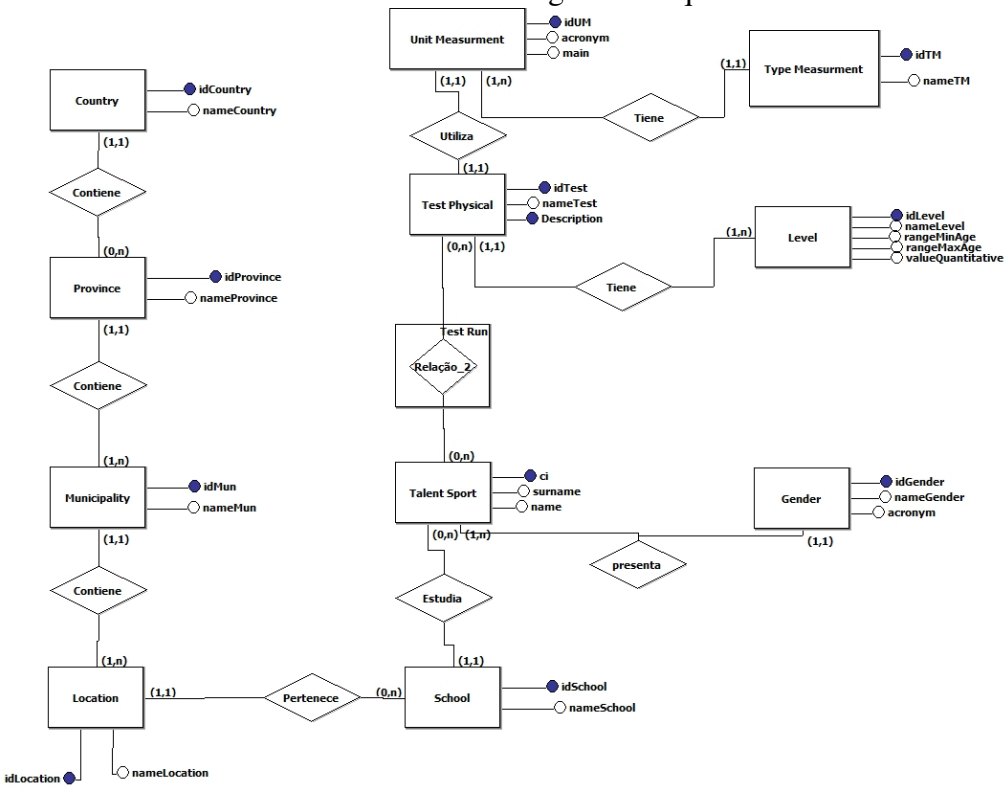


- accescontrol: agrupa los modelos, la lógica y validaciones referentes a la app
- aplication: agrupa los modelos, la lógica y validaciones referentes a la app
- core: agrupa los modelos, la lógica y validaciones referentes a la app
- country: agrupa los modelos, la lógica y validaciones referentes a la app country
- extra: agrupa los modelos, la lógica y validaciones referentes a la app
- gender: agrupa los modelos, la lógica y validaciones referentes a la app gender
- medias: agrupa los modelos, la lógica y validaciones referentes a la app medias
- method evauation: agrupa los modelos, la lógica y validaciones referentes a la app method evauation
- school: agrupa los modelos, la lógica y validaciones referentes a la app school
- sex: agrupa los modelos, la lógica y validaciones referentes a la app sex
- talent sport: agrupa los modelos, la lógica y validaciones referentes a la app talent sport

- test physical: agrupa los modelos, la lógica y validaciones referentes a la app test physical
- tamplates: contiene todas las vistas del proyecto
- unit measument: agrupa los modelos, la lógica y validaciones referentes a la app unit measument
- softeef2.0: la carpeta contenedora del proyecto

2.9 Diseño de base de datos

Para el desarrollo del sistema se modeló el siguiente esquema de base de datos:



Descripción de las entidades del modelo de datos:

1. *talentSport*: almacena la información de todos los talentos deportivos almacenados en la base de datos, carnet de identidad, nombre y apellidos.
2. *gender*: alamcena la información en la base de datos del sexo, nombre de sexo y acrónimo.

3. *province*: almacena la información de todas las provincias almacenadas en la base de datos, nombre.

4. *municipality*:almacena la información de todos los municipios almacenados en la base de datos, nombre.

5. *location*: alamcena la información en la base de datos de localidades, nombre.

6. *school*: alamcena la información en la base de datos de escuelas, nombre.

7. *unitMeasurement*:almacena la información de todas las unidades de medidas almacenados en la base de datos, nombre y acrónimo.

8. *typeMeasurement*: almacena la información de todas las magnitudes física, nombre de la magnitud.

9. *testPhysical*:almacena la información de todas las pruebas físicas almacenados en la base de datos, nombre de la prueba y descripción.

10. *level*: alamcena la información en la base de datos de niveles, nombre, rango mínimo de edad, rango máximo de edad y valor cuantitativo.

2.10 Tarjetas CRC

Las tarjetas CRC (Class Responsibility Collaborator) es una actividad en grupo de modelado orientado a objetos en la cual el equipo puede manifestar y debatir ideas acerca del diseño de un sistema. Hace especial énfasis en la simplicidad, comunicación y límites de un sistema. Se suele utilizar en las primeras fases del desarrollo de una historia como paso previo a la implementación o escritura de un esquema UML.(Márquez Ramos, Aparicio Reytor y Armas Rodríguez, 2014)

El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan en la parte izquierda, y las clases que se implican en cada responsabilidad en la parte derecha. Clase: es cualquier persona, evento, concepto, pantalla o reporte, Responsabilidades: las responsabilidades de una clase son las entidades que conoce y las que realizan sus atributos y métodos, Colaboradoras: las colaboradoras de una clase son las demás clases

con las que trabaja en conjunto para llevar a cabo sus responsabilidades. A continuación se muestra unas de las tarjetas CRC obtenidas del sistema:

Nombre de Clase: UpdateViewSchool ***Data***	
Superclase: View *Object*	**Subclases:** *Measurement*,*Serie*,*Variable*
Responsibilidades	**Colaboradoras**
Controlan las peticiones desde la vista de actualizar escuelas.	UpdateSchoolForm, super,School

2.11 Tarea de Ingeniería

Las tareas de ingeniería comprenden un conjunto de actividades fundamentales. Esto incluye la comprensión de los requisitos del cliente, la planificación del proyecto, el diseño arquitectónico del sistema, la codificación del software, las pruebas para asegurar su funcionalidad y calidad, así como la implementación y el mantenimiento continuo. La ingeniería de software implica también la gestión eficiente de recursos, el seguimiento de plazos y costos, la documentación adecuada de cada etapa del desarrollo, y la colaboración estrecha con equipos multidisciplinarios para garantizar que el software producido cumpla con las expectativas del cliente y las necesidades del usuario final. Estas tareas de ingeniería son fundamentales para el éxito del proceso de desarrollo, asegurando la creación de software robusto, confiable y que cumple con los estándares de calidad establecidos.(Carvajal Rojas, 2005)

En la siguiente tabla se muestra el formato utilizado para la confección de las tareas de ingeniería:

Tabla 2.8: Ejemplo de Tarea de Ingeniería.

Tarea de Ingeniería	
Número tarea: 6	**Número de Historia de Usuario:** 4.04
Nombre tarea: Eliminar atleta.	
Continúa en la siguiente página	

Tabla 2.8 Continuación de la página anterior

Tipo de tarea: Desarrollo	**Puntos estimados:** 0.1
Fecha inicio: 2 de octubre del 2023	**Fecha fin:** 2 de octubre del 2023
Programador responsable: Javier Alejandro González Muñiz	
Descripción: El sistema debe brindar la funcionalidad de eliminar un atleta de la tablas.	

Los campos de la tarjeta de las Tareas de Ingeniería reflejan lo siguiente:

- **Número tarea:** Representa el número por el que se identifica a la tarea. Cada tarea tiene un único número que la identifica.
- **Número Historia de Usuario:** Es el número de la Historia de Usuario a la que responde la tarea.
- **Nombre de tarea:** Define el nombre o funcionalidad concreta a la que se dedica la tarea, debe estar expresado en forma infinitiva.
- **Tipo de tarea:** Información del tipo de tarea a realizar, la misma puede ser:
  - **Desarrollo:** Tarea que se realizará por primera vez.
  - **Corrección:** Tarea que se realiza a partir de una anterior que no se realizó correctamente, es decir, que no pasó los casos de prueba satisfactoriamente.
  - **Mejora:** Tarea que se realiza a partir de una anterior incorporándole nuevos requerimientos.
  - **Otra:** Tarea que no corresponde con una de las anteriores, en este caso es necesario especificar el tipo de tarea o realizar una descripción más profunda de esta.
- **Puntos estimados:** Tiempo de duración de la tarea. El tiempo estimado es reflejado en días. La suma de los puntos estimados de las tareas de ingeniería de una Historia de Usuario no puede superar la cantidad de puntos estimados definidos para la Historia de Usuario.

- **Fecha inicial:** Fecha en la que se inicia el desarrollo de la tarea de ingeniería.
- **Fecha final:** Fecha en la que se concluye el desarrollo de la tarea de ingeniería.
- **Programador responsable:** Nombre del responsable de la realización de la tarea.
- **Descripción:** Es una breve descripción sobre lo que la tarea debe hacer o resolver.

2.12 Pruebas

La metodología XP enfatiza en la realización de pruebas a lo largo de todo el desarrollo del software, con el fin de lograr un producto con calidad, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su corrección. En este proceso no solo participa el desarrollador, también es importante la colaboración del cliente, sobre todo en las pruebas de aceptación. En XP las pruebas se dividen en dos grupos: pruebas unitarias encargadas de verificar el código y pruebas de aceptación orientadas a probar las funcionalidades del sistema

2.12.1 Casos de prueba

Los casos de prueba son evidencias de pruebas funcionales o unitarias que se realizan al sistema para comprobar su funcionamiento. Las pruebas funcionales son validaciones escritas desde la perspectiva del cliente, y las pruebas unitarias son validaciones desde la perspectiva del programador. El objetivo general es tener una forma para decirle al cliente que la Historia de Usuario está lista. Las pruebas funcionales o pruebas de aceptación, son las más importantes, ya que representan la medida de satisfacción del cliente para una funcionalidad que el sistema debe tener. Los casos de pruebas fueron definidos para cada Historia de Usuario establecida. Se comprueba el funcionamiento de cada una de las funcionalidades implementadas que responden a la misma. El formato utilizado para la confección de casos de pruebas se muestra a continuación en la tabla.

Tabla 2.9: Ejemplo de los casos de prueba.

Caso de prueba de aceptación	
Código: PA01_HU01	**Historia de Usuario:** HU01
Nombre: Auntenticar al sistema con errores I	
Descripción: Se intenterá autenticar al sistema dejando los campos de usuario y contraseña vacío.	
Condiciones de ejecución: El sistema no debe haber iniciado sesión o tenerla cerrada en el servidor.	
Pasos de ejecución: Se ingresa a la pantalla de autenticación mediante la url en el navegador. Una vez que aparezca la vista inicial, se presiona el botón Autenticar del cuadro de diálogo con los campos usuario y clave vacíos.	
Resultados esperados: Se debe visualizar un cuadro notificación que alerte de que existen campos vacíos. Se debe señalar de forma gráfica aquellas cajas de texto que están vacías.	
Evaluación de la prueba: Satisfactoria	

Campos del caso de prueba

- *Código:* Identificador del caso de prueba. Dividido en dos partes. La primera representa la inicial del artefacto y la segunda representa el número con que se identifica la prueba.
- *Historia de Usuario:* Es el número de la Historia de Usuario a la que responde el caso de prueba.
- *Descripción:* Es una breve descripción del propósito de la prueba.
- *Condiciones de ejecución:* Condiciones especiales que deben tenerse en cuenta para ejecutar el caso de prueba.

- *Entradas / pasos de ejecución:* Entradas o funciones que deben ejecutarse para realizar el caso de prueba.
- *Resultado esperado:* Salida u objetivo que debe cumplir la funcionalidad a la que se le realiza el caso de prueba.
- *Evaluación:* Evaluación de éxito del caso de prueba. Prueba satisfactoria en caso de éxito o prueba insatisfactoria en caso de fallo.

Los casos de prueba son agregados a los artefactos de entrega que se realiza al cliente al terminar cada fase o iteración del proyecto. Las Historias de Usuario con evaluación insatisfactoria, serán corregidas en la próxima iteración a partir de nuevas tareas de ingeniería. Los casos de pruebas realizados al sistema se encuentran adjuntos en los anexos del documento.

2.12.2 Pruebas de aceptación

Las pruebas de aceptación son las especificaciones para el comportamiento deseado y la funcionalidad de un sistema. Muestra por una Historia de Usuario dada, cómo el sistema se encarga de ciertas condiciones y con qué tipo de resultados. Los clientes junto a un miembro del equipo de desarrollo son los responsables de verificar que los resultados de estas pruebas sean los correctos para así tomar decisiones acerca de las mismas. Una Historia de Usuario no se puede considerar terminada hasta que no pase las pruebas de aceptación. Es recomendable publicar los resultados de las pruebas de aceptación, para que todo el equipo de desarrollo esté al tanto de esta información. Al sistema además de las pruebas de funcionalidad, se le realizaron Pruebas de Regresión para comprobar que las no conformidades detectadas habían sido resueltas y que no se había afectado otras funcionalidades.(Mascheroni, Cogliolo e Irrazabal, 2016)

2.12.3 Pruebas de compatibilidad

Que un sitio web sea compatible con todos los navegadores significa que se vea igual (o muy similar) en todos ellos. Algunos autores consideran como suficiente si el sitio puede

ser percibido por el usuario con las mismas características, en los navegadores más importantes, como Internet Explorer, Firefox, Chrome, Opera, Safari y Mozilla. Las Pruebas de Compatibilidad, son pruebas que se realizan en una aplicación determinada comprobando su compatibilidad con todos los navegadores de Internet y sistemas operativos del mercado. Para ello, existen diferentes técnicas:()

- Verificación del cumplimiento de estándares (como, por ejemplo, W3C o ECMA): Consiste en analizar los componentes gráficos del sitio web en diferentes navegadores, para verificar que sigan los lineamientos y especificaciones de los estándares. Esta técnica se utilza preferentemente en etapas intermedias del proceso de desarrollo, teniéndose que el diseño real se estará llevando a cabo con la consideración de los estándares que correspondan.

- Pruebas de interfaz de usuario (UI): Es la más común de las técnicas. Puede ser realizada de forma manual o con software especializado (pruebas automatizadas). El objetivo de este tipo de prueba es revisar el contenido visual del sitio Web a través de la navegación de sus páginas en los diferentes navegadores.

- Análisis del modelo de objetos del documento (DOM): Es una técnica dinámica que consiste en comparar el comportamiento de una aplicación web en diferentes navegadores, identificando las diferencias como defectos. La comparación se realiza combinando analisis estrucural de la información en el DOM junto con un análisis visual de la página en cuestión.

- Comparación de Imágenes: Ésta técnica se basa en tomar una captura de pantalla del sitio en un tipo de navegador, y compararla con otra captura del sitio en otro navegador diferente del primero. Si ambas imágenes coinciden, entonces el sitio será compatible entre ambos navegadores.

2.12.4 Pruebas de usabilidad

Las pruebas de usabilidad son un servicio de aseguramiento de calidad que consiste en invitar a profesionales, cuyo perfil se adapta al de su público objetivo, a probar el producto

y proporcionar comentarios valiosos sobre su facilidad de uso y eficiencia. El objetivo principal de las pruebas de usabilidad es identificar los problemas de usabilidad, recolectar comentarios pertinentes y mejorar la satisfacción de sus usuarios (García, 2015).
Una de las pruebas de usabilidad más frecuentemente realizada, son las *heurísticas de Nielsen* (Nielsen, 1995). Nielsen, tras analizar varios artículos y libros plantea diez heurísticas que permiten evaluar la usabilidad de un producto. Dichas heurísticas plantea:

1. Visibilidad del estado del sistema. El sistema siempre debe mantener a los usuarios informados sobre lo que está pasando, a través de una retroalimentación adecuada dentro de un tiempo razonable.

2. Correspondencia entre el sistema y el mundo real. El sistema debe hablar el idioma de los usuarios, con palabras, frases y conceptos familiares para el usuario, en lugar de términos orientados al sistema. Haciendo que la información aparezca en un orden natural y lógico.

3. Control del usuario y libertad. Los usuarios suelen elegir funciones del sistema por error y necesitarán una *salida de emergencia* claramente marcada para dejar el estado no deseado sin tener que pasar por un diálogo extendido. La posibilidad de deshacer y rehacer acciones.

4. Consistencia y estándares. Los usuarios no deberían preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Siga las convenciones de la plataforma.

5. Prevención de errores. Incluso mejor que los buenos mensajes de error, es un diseño cuidadoso que impide que se produzca un problema en primer lugar. Elimine las condiciones propensas a errores o compruébelo y presente a los usuarios una opción de confirmación antes de comprometerse con la acción.

6. Reconocimiento en lugar de recordar. Minimice la carga de memoria del usuario haciendo visibles los objetos, las acciones y las opciones. El usuario no debe tener que recordar la información de una parte del diálogo a otra. Las instrucciones de uso del sistema deben ser visibles o fácilmente recuperables cuando sea apropiado.

7. Flexibilidad y eficiencia de uso. Los aceleradores no vistos por el usuario principiante pueden a menudo acelerar la interacción para el usuario experto, de manera que el sistema pueda atender a usuarios inexpertos y experimentados. Permite a los usuarios adaptar acciones frecuentes.

8. Diseño estético y minimalista. Los diálogos no deben contener información irrelevante o raramente necesaria. Cada unidad extra de información compite con las unidades relevantes de información y disminuye la visibilidad relativa.

9. Ayuda a los usuarios a reconocer, diagnosticar y recuperar errores. Los mensajes de error deben expresarse en lenguaje sencillo (sin códigos), indicar con precisión el problema y sugerir constructivamente una solución.

10. Ayuda y documentación. A pesar de que es mejor si el sistema puede utilizarse sin documentación, puede ser necesario proporcionar ayuda y documentación. Cualquier información de este tipo debe ser fácil de buscar, centrada en la tarea del usuario, enumerar los pasos concretos que se deben llevar a cabo, y no ser demasiado grande.

2.12.5 Pruebas de satisfacción de usuarios

Otro factor a analizar para saber si el sistema cumple con los requisitos necesarios de cara al cliente es realizar pruebas para determinar el agrado del usuario. La población utilizada para estas pruebas constaba de un equipo de 6 personas que enviaban comentarios y sugerencias para el sistema. Para este cometido se ha realizado una encuesta a estos usuarios, donde se preguntaban sobre las siguientes cuestiones:

1. El sistema es fácil de entender para el usuario.

2. La interfaz del sistema es atractiva y amigable.

3. El tiempo de respuesta es correcto bajo ciertas condiciones.

4. En términos generales, el sistema cumple su cometido y con buen rendimiento.

5. ¿Qué es lo que más le ha gustado de la aplicación?.

- Interfaz.
- Facilidad de uso.
- Funcionalidades.
- Tiempo de respuesta / Rendimiento

2.13 Conclusiones parciales del cappítulo

En el capítulo se exploran aspectos clave para el desarrollo del sistema, centrándose en el modelo de dominio, los requisitos, y los artefactos que conforman la metodología de desarrollo. Se destaca el uso del patrón arquitectónico Modelo-Vista-Template (MVT) para guiar la arquitectura del sistema, lo cual permitió una clara separación entre la capa de datos, la lógica de negocio y la presentación al usuario. Además, se profundiza en los patrones de diseño empleados, resaltando su importancia para definir una arquitectura más sólida y eficiente. En conjunto, estos elementos proporcionaron una base sólida para el desarrollo del sistema, garantizando una arquitectura bien definida y adaptada a los objetivos planteados.

Capítulo 3: Implementación y prueba

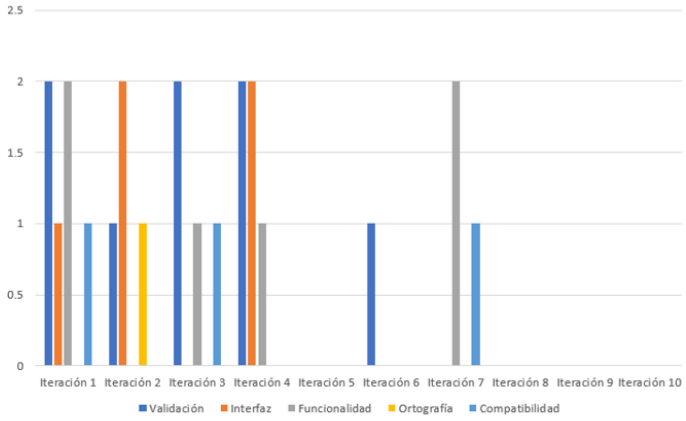
Introducción

En este capítulo se aborda los resultados de las pruebas definidas en el capítul anterior, se muestran algunas de las interfaces y funcionalidades del sistema.

3.1 Resultados de las pruebas

3.1.1 Pruebas de aceptación

Como resultado de las pruebas de aceptación se detectaron un total de 21 no conformidades. A medida que se avanzó en las iteraciones se disminuyeron los números de no conformidades hasta ser cero. De esta manera el sistema queda listo para ser utilizado. En la siguiente figura se resumen las no conformidades detectadas en cada una de las iteraciones:



3.1.2 Pruebas de compatibilidad

El sistema fue probado en los navegadores Opera, Mozilla Firefox, Google Chrome y Microsoft Edge, obteniéndose resultados positivos. Los estándares de la W3C (Google) utilizados no sufren ninguna variación en los diferentes tipos de navegadores, ni en los colores, ni en las animaciones. Las funcionalidades del sistema tampoco mostraron ningún error.

3.1.3 Pruebas de usabilidad

Para llegar a uno o varios resultados en las pruebas de usuabilidad, se debe partir del cumplimiento en el sistema de las diez heurísticas que permiten evaluar la usabilidad de un producto según Nielsen:

1. Este primer aspecto es cumplido por el sistema el cual muestra en todo momento en las tablas los listados de los diferentes datos actualizados, así como las notificaciones.

2. Los mensajes de error aparecen en el momento que el usuario realiza alguna operación tanto si se está autenticando o adicionando, modificando o eliminando algún dato en el sistema, estos son concisos y no utilizan palabras técnicas o ambiguas que puedan confundir al usuario cuando sucede algún error.

3. En el sistema existen botones para cancelar las diferentes operaciones, un botón atrás, la ruta de migas, un panel de navegación, con lo cual, si el usuario quiere deshacer su acción puede hacerlo sin ningún inconveniente y en todo momento.

4. El sistema utiliza componentes estándares a los que los usuarios ya están acostumbrados al interactuar con ellos y a su vez estos responden de manera intuitiva a posibles interacciones. Las palabras utilizadas en botones, etiquetas, títulos y otros componentes guardan relación con la acción asociada.

5. El sistema utiliza cuadros de diálogos en los que el usuario tiene que ingresar datos en varios campos de texto, los cuales pueden contener errores y el sistema debe ser capaz de señalarlos. Para evitar esto, en el momento donde el usuario envíe los datos al servidor, si estos son incorrectos, serán señalados en dicho cuadro de diálogo, además, existirán restricciones de entrada de datos en algunos campos de texto según el tipo de dato que se debe introducir.

6. El sistema cuenta con varios componentes interactivos, todos están visibles y situados de forma que se minimice el riesgo de que el usuario pueda realizar acciones erróneas accidentalmente. En todo momento, gracias a la ruta de migas, se indica dónde se

encuentra el usuario y la información necesaria en dependencia de la vista en la que se encuentre.

7. Este aspecto no se aprecia, ya que el sistema no hace uso de aceleradores, accesos directos o atajos del teclado.

8. El diseño de la aplicación está basado en los principios de estándares de Google, de esta forma, se garantiza que el diseño sea minimalista y acorde con la plataforma. En cuanto a la estética se tuvo en cuenta los tamaños de las fuentes utilizadas en correspondencia con la función que el texto desempeña, los controles alineados y uniformes que se adapten de igual forma para cualquier navegador.

9. Los mensajes de error son precisos sin llegar a una formalidad que desinterese a los usuarios, pero tampoco con una informalidad que los incomode; e indicando qué hacer para solucionar el problema.

10. Se ofrecen los manuales de instalación y de usuario que servirán de guía tanto para el despliegue como para el trabajo en el sistema.

3.1.4 Pruebas de satisfacción de usuarios

Para las cuatro primeras pruebas de satisfacción se les solicitó a los usuarios involucrados en estas pruebas que dieran una valoración de 1 a 5. Donde 1 significa estar en desacuerdo y 5 si esta completamente de acuerdo.

El sistema es fácil de entender para el usuario

Tabla 3.1: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	2	0.00
Valoración de 2	2	0.00
Valoración de 3	0	0.00

Valoración de 4	4	40.00
Valoración de 5	0	60.00

La interfaz del sistema es atractiva y amigable

Tabla 3.2: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	0	0.00
Valoración de 2	0	0.00
Valoración de 3	0	0.00
Valoración de 4	0	0.00
Valoración de 5	5	100.00

Como se puede apreciar, los usuarios consideran el diseño de la interfaz de la aplicación atractiva y amigable.

En términos generales, el sistema cumple su cometido y con buen rendimiento

Tabla 3.3: Resultados de la pregunta 1 de las pruebas de satisfacción de usuarios **Fuente:** Elaboración propia

Valoración	Votos	Por ciento
Valoración de 1	0	0.00
Valoración de 2	0	0.00
Valoración de 3	0	0.00
Valoración de 4	2	40.00
Valoración de 5	3	60.00

Como se puede apreciar, los usuarios están satisfechos con las funcionalidades que brinda el sistema. Se podría decir que se ha obtenido el rendimiento y la cantidad de funcionalidades

que todos los usuarios esperaban. Por lo tanto podemos concluir que el sistema cumple con los requisitos no funcionales expuestos en la fase de exploración.

3.2 Vistas del sistema

Se muestran a continuación algunas de las vistas del sistemas así como sus funcionalidades.

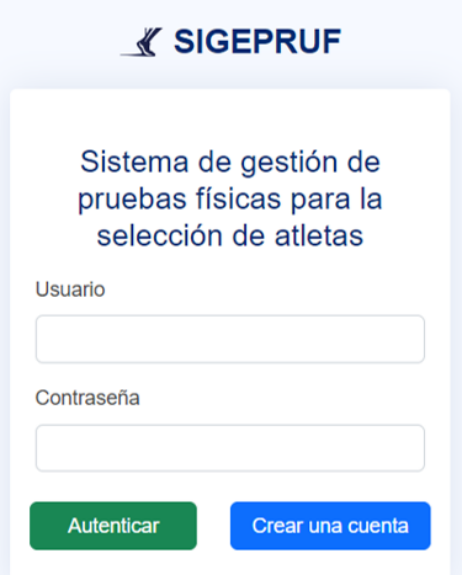


Figura 3.1: Vista de autenticación al sistema, muestra el nombre y logo del sistema. Una vez el usuario ha sido insertado al sistema, este podrá acceder con el rol corrspondiente

3.3 Conclusiones parciales del capítulo

Se muestran los resultados de las pruebas defnidas en el cápitulo anterior que se someterá el software, según la metodología de software escogida en conjunto con otros tipos de pruebas para la validación del sistema desarrollado y gracias a estas se detectaron y se solucionaron las fallas existentes en el sistema. Se muestran algunas vistas del sistemas y sus funcionalidades. Por tanto, el sistema web SIGEPRUF está listo para ser desplegado y utilizado por el Grupo de Creación Artística Vigía.

Figura 3.2: Dashboard o menú de navegación, consiste en la navegación del sitio. Se encuentra definida toda la navegacion de la aplicación, donde el usuario puede acceder a todos los modulos

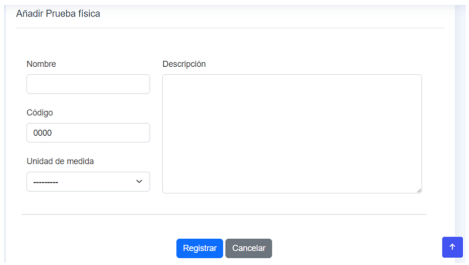


Figura 3.3: Registrar pruebas físicas al sistema. Para ello el usuario deberá cumplir con los campos requeridos y los datos cumplir con la validación definida

Conclusiones

Con la realización de la presente investigación se establecieron los fundamentos teórico-metodológicos de los principales elementos que componen la gestión de pruebas físicas en el proceso de selcción de atletas en la provincia de Matanzas. Se definió una metodología de desarrollo para el sistema pruebas físicas en el proceso de selcción de atletas en la provincia de Matanzas de acuerdo al estado del arte de las principales tecnologías, metodologías y herramientas para el desarrollo de este tipo de sistemas, se especificaron los requisitos funcionales y no funcionales, lo que permitió identificar las funcionalidades que dan respuesta a las necesidades del usuario.

Se implementó el sistema web SIGEPRUF según los presupuestos y requisitos previamente definidos, haciendo uso de herramientas de software libre y código abierto. Además, se validó su funcionamiento mediante la realización de las pruebas necesarias, según la metodología XP y posteriormente se realizó el análisis de los resultados que arrojaron las mismas.

Con la implementación del software, se evidencia una mejora en la gestión de pruebas físicas en el proceso de selcción de atletas en la provincia de Matanzas. Cumpliendo con una organización y centralización de los datos, siendo este uno de los principales problemas definidos en el inicio de la investigación.

Referencias Bibliográficas

Alvarez, M. A. (2010). "Manual de jQuery". En: *Recuperado de http://dmaspv. com/files/page/0704201118 nual % 20de % 20jquery % 20en % 20pdf % 20desarrolloweb-com. pdf*.

Andreu, J. (2011). *Gestión de servidores web (Servicios en red)*. Editex.

Aranguren Peraza, G. (2007). "La investigación-acción sistematizadora como estrategia de intervención y formación del docente en su rol de investigador". En: *Revista de pedagogía* 28.82, págs. 173-195.

Beck, K., M. Hendrickson y M. Fowler (2001). *Planning extreme programming*. Addison-Wesley Professional.

Carvajal Rojas, J. H. (2005). "Metodología de diseño mecatrónico de robots". En: *Épsilon* 1.4, págs. 91-101.

Christudas, B. y B. Christudas (2019). *MySQL*. Springer.

Cruz Vélez, P. E. De la, D. E. B. Loayza, M. R. Huamán et al. (2005). "Documentación de Proyectos Con UML 2". En: *Revista de investigación de Sistemas e Informática* 7.1, págs. 27-36.

Espinosa-Hurtado, R. (2021). "Análisis comparativo para la evaluación de frameworks usados en el desarrollo de aplicaciones web". En: *CEDAMAZ* 11.2, págs. 133-141.

Foundation, M. (2022). *CSS: Cascading Style Sheets*. Consultado: 2022-06-24. URL: https://developer.mozilla.org/en-US/docs/Web/CSS.

García, X. (2015). *UF1843 - Aplicaciones técnicas de usabilidad y accesibilidad en el entorno cliente*. Ediciones Paraninfo, S.A. ISBN: 9788428397810.

García Gómez, I. (2020). "Sistema de gestión de atletas a partir de la arquitectura de microservicios". En.

Gómez García, D. E. (2018). "Desarrollo del sistema de requisiciones para la empresa hidroeléctrica Abanico SA Aplicando el entorno de programación Node. js". B.S. thesis. Escuela Superior Politécnica de Chimborazo.

Jiménez, J. (2016). *UF2406 - El cliclo de vida del desarrollo de aplicaciones*. Editorial Elearning, S.L.

Jiménez-Builes, J. A., D. L. Ramírez-Bedoya y J. W. Branch-Bedoya (2019). "Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP". En: *Revista politécnica* 15.30, págs. 55-69.

Johnson, B. (2012). *Professional visual studio 2012*. John Wiley & Sons.

Leal, E. T., C. M. Sosa y D. A. T. Leal (2012). "Revisión de los sistemas de control de versiones utilizados en el desarrollo de software". En: *Ingenierías USBMed* 3.1, págs. 74-81.

Letelier, P. (2006). "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)". En.

Lopez-Pellicer, F. J. et al. (2015). "GitHub como herramienta docente". En: *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática*. Universitat Oberta La Salle, págs. 66-73.

Maida, E. G. y J. Pacienzia (2015). "Metodologías de desarrollo de software". En.

Márquez Ramos, A., M. Aparicio Reytor y N. de Armas Rodríguez (2014). "MÓDULO PARA LA EXTRACCIÓN DE CARACTERÍSTICAS DE LAS FUENTES DE INFORMACIÓN PARA EL SISTEMA DE VIGILANCIA TECNOLÓGICA." B.S. thesis.

Mascheroni, M. A., M. K. Cogliolo y E. Irrazabal (2016). "Automatización de pruebas de compatibilidad web en un entorno de desarrollo continuo de software". En: *Simposio Argentino de Ingeniería de Software (ASSE 2016)-JAIIO 45 (Tres de Febrero, 2016).*

Molina Hernández, Y., A. Granda Dihigo y A. Velázquez Cintra (2019). "Los requisitos no funcionales de software. Una estrategia para su desarrollo en el Centro de Informática Médica". En: *Revista Cubana de Ciencias Informáticas* 13.2, págs. 77-90.

Montoya, C. E. G., C. A. C. Uribe y L. E. S. Rodríguez (2013). "Seguridad en la configuración del servidor web Apache". En: *Inge Cuc* 9.2, págs. 31-38.

Muñoz, V. (2012). *Aprendiendo a programar paso a paso con C*. Vicente Javier Eslava Muñoz. ISBN: 9788468610610.

Navarrete, T. (2006). "El lenguaje JavaScript". En: *Asignatura: Fundamentos de Cartografia i SIG*.

Nielsen, J. (1995). "10 usability heuristics for user interface design". En: *Nielsen Norman Group* 1.1.

Oscar, S. (2013). *Visual Paradigm for Uml*. International Book Market Service Limited. ISBN: 9786139166534.

Paradigm, V. (2022). *Visual Paradigm Product Overview*. Consultado: 2022-06-24. URL: https://www.visual-paradigm.com/support/documents/vpuserguide/12/13/5963_visualparadi.html.

Parra, J. (2018). *UF2218 - Desarrollo de un CMS*. Editorial Elearning, S.L.

Ríos, J. R. M. et al. (2016). "Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python". En: *Revista latinoamericana de Ingenieria de Software* 4.4, págs. 201-207.

Salazar Chang, I. R. (2015). "Diseño de un sistema de Gestión de la Calidad Total en el ámbito del deporte. Modelo MEXD de Excelencia Deportiva". En.

Sánchez Muguercia, N. (2013). *Aplicación de escritorio del Sistema de Entrega Digital*. B.S. thesis.

Silva, A et al. (2018). "Utilidad del Lenguaje Unificado de Modelado (UML) en el desarrollo de software profesional dentro del sector empresarial y educativo". En: *Ciencia-Cierta revista de divulgación científica* 56.

Toro, A. D. y B. B. Jiménez (2000). "Metodología para la elicitación de requisitos de sistemas software". En: *Informe Técnico LSI-2000-10. Facultad de Informática y Estadística Universidad de Sevilla*.

Troya Matos, J. J. (2020). "Estudio de factibilidad para el diseño de un sistema de gestión y seguimiento de los entrenamientos para los deportistas de alto de rendimiento en la Provincia de Los Ríos". B.S. thesis. BABAHOYO: UTB, 2020.

VALENCIA, H. P. (2013). “SISTEMA INFORMÁTICO PARA LA ADMINISTRACIÓN DE EXPEDIENTES DEPORTIVOS Y SEGUIMIENTO DE PLANES DE ENTRENAMIENTO DEL INSTITUTO NACIONAL DE LOS DEPORTES”. En.

Vázquez, J. I. C. et al. (2018). “Uso de herramientas CASE para la gestión de proyectos de software”. En: *Pistas Educativas* 35.110.

Vázquez-Ingelmo, A y F. García-Peñalvo (2019). “Modelo de Dominio”. En.

Vega, G. E. et al. (2019). “Una comparación de rendimiento entre bases de datos NoSQL: MongoDB y ArangoDB”. En: *Tecnología en Marcha* 32.9, págs. 5-15.

Vértice, E. (2009). *Diseño básico de páginas web en HTML*. Editorial Vértice.