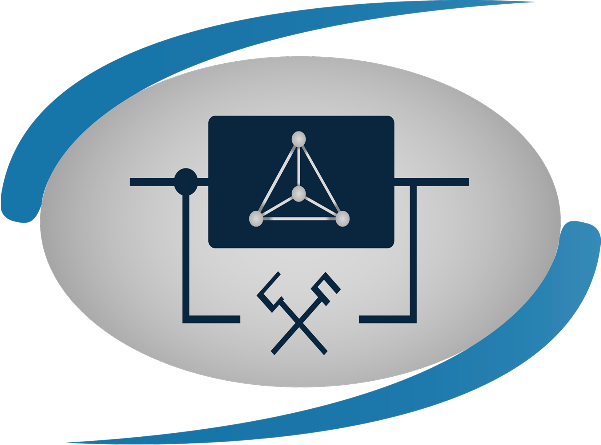


UNIVERSIDAD TÉCNICA DE ORURO

FACULTAD NACIONAL DE INGENIERÍA

INGENIERÍA SISTEMAS



PROYECTO SIS – 2420 A “ACTUALIZACIÓN TECNOLÓGICA”

**SISTEMA DE INFORMACIÓN PARA LA ADMINISTRACIÓN DE UNA CANCHA DE WALLY (ANDYNA)**

**Docente:** Ing. Saul Mamani Mamani

**Team:** Unidos por el Objetivo

**Estudiante:** Univ. Mamani Colque Ronald “A”

Univ. Varela Poma Angel Dario “B”

Univ. Roque Chaparro David Jose “B”

Univ. Villca Flores Wilfredo “A”

Oruro – Bolivia

2023

INDICE

[1 INTRODUCCIÓN 1](#_Toc153122894)

[1.1 Presentación 1](#_Toc153122895)

[1.2 Planteamiento del problema 1](#_Toc153122896)

[1.3 Objetivos 1](#_Toc153122897)

[1.3.1 Objetivo general 1](#_Toc153122898)

[1.3.2 Objetivos específicos 2](#_Toc153122899)

[1.4 Alcances 2](#_Toc153122900)

[1.5 Ingeniería del proyecto 2](#_Toc153122901)

[2 MARCO TEÓRICO 4](#_Toc153122902)

[2.1 Algoritmo 4](#_Toc153122903)

[2.2 Software 4](#_Toc153122904)

[2.3 Sistema de información 4](#_Toc153122905)

[2.4 Scrum 4](#_Toc153122906)

[2.5 Modelado 4](#_Toc153122907)

[2.6 UML: Lenguaje de Modelado Unificado 4](#_Toc153122908)

[2.6.1 Diagrama de casos de uso 4](#_Toc153122909)

[2.6.2 Diagrama de clases 4](#_Toc153122910)

[2.6.3 Diagrama de despliegue 4](#_Toc153122911)

[2.7 .Net Framework 4](#_Toc153122912)

[2.8 C#.NET Core 4](#_Toc153122913)

[2.9 ASP.NET Core 4](#_Toc153122914)

[2.10 SQL Server 4](#_Toc153122915)

[2.11 Arquitectura MVC 4](#_Toc153122916)

[2.12 Enterprise Architect 4](#_Toc153122917)

[2.13 Balsamiq 4](#_Toc153122918)

[3 MARCO PROPOSITIVO 5](#_Toc153122919)

[3.1 Diagrama de contexto 5](#_Toc153122920)

[3.2 Equipo Scrum y los Stakeholders 5](#_Toc153122921)

[3.3 Historias de Usuario 5](#_Toc153122922)

[3.4 Producto Backlog 7](#_Toc153122923)

[3.5 Release Planning 7](#_Toc153122924)

[3.6 Diagrama de casos de uso del sistema 8](#_Toc153122925)

[3.7 Diagrama de clases 8](#_Toc153122926)

[3.8 Modelo Relacional de la Base de Datos 9](#_Toc153122927)

[3.9 Primer Sprint 10](#_Toc153122928)

[3.9.1 Sprint backlog 10](#_Toc153122929)

[3.9.2 Diseño de interfaces 10](#_Toc153122930)

[3.9.3 Sprint review 11](#_Toc153122931)

[3.9.4 Sprint retrospective 12](#_Toc153122932)

[3.10 Segundo Sprint 13](#_Toc153122933)

[3.10.1 Sprint backlog 13](#_Toc153122934)

[3.10.2 Diseño de interfaces 13](#_Toc153122935)

[3.10.3 Sprint review 13](#_Toc153122936)

[3.10.4 Sprint retrospective 13](#_Toc153122937)

[3.11 Tercer Sprint 14](#_Toc153122938)

[3.11.1 Sprint backlog 14](#_Toc153122939)

[3.11.2 Diseño de interfaces 14](#_Toc153122940)

[3.11.3 Sprint review 14](#_Toc153122941)

[3.11.4 Sprint retrospective 14](#_Toc153122942)

[4 CONCLUSIONES Y RECOMENDACIONES 15](#_Toc153122943)

[BIBLIOGRAFIA 16](#_Toc153122944)

[ANEXOS 17](#_Toc153122945)

# **1 INTRODUCCIÓN**

## **Presentación**

Una de las motivaciones para realizar este proyecto es que toda persona necesita realizar actividades físicas para mantener una mejor salud. Los deportes han sido desde sus orígenes parte fundamental en el desarrollo físico de la persona y para cultivar momentos de entretenimiento y también incentivando la competencia.

El presente sistema cuenta con una amplia gama de funcionalidades los cuales ayudan tanto al dueño en la facilitación de la administración, el registro y el control del alquiler de las canchas, como al cliente en el registro de reservación de su cancha o canchas, dando a los usuarios la facilidad y la comodidad necesaria para que de esta forma pueda disfrutar de una experiencia reconfortante y agradable.

En el sistema también se puede realizar las reservaciones de forma online, se podrá seleccionar la fecha, el horario y el número de cancha que desee alquilar, siendo más factible la organización de los partidos que se jugaran y disfrutaran, en las canchas respectivas.

## **1.2 Planteamiento del problema**

En la actualidad, el proceso de reserva de canchas de wally suele ser complicado y poco práctico, con sistemas obsoletos basados en llamadas telefónicas o visitas presenciales al lugar. Esto puede generar inconvenientes como la falta de disponibilidad de horarios deseados, la dificultad para realizar reservas de último minuto y la limitada visibilidad de la disponibilidad de las canchas. Por lo tanto, se plantea la necesidad de desarrollar un sistema automatizado que simplifique y agilice el proceso de reserva de canchas de wally, mejorando así la experiencia de los usuarios.

## **1.3 Objetivos**

El objetivo principal de este estudio es mejorar el proceso de alquiler de canchas de Wally para garantizar una experiencia satisfactoria para los usuarios y una gestión eficiente para los propietarios de las instalaciones.

### **Objetivo general**

Desarrollar un sistema de información, para administrar, registrar y controlar el alquiler y la reservación de las canchas de Wally.

**1.3.2 Objetivos específicos**

* Identificar historias de usuario para capturar los requerimientos funcionales que guíen el desarrollo del sistema.
* Realizar diagramas de casos de uso, clases y secuencias con UML, para modelar el análisis y el diseño del sistema.
* Diseñar una base de datos centralizada, para almacenar toda la información necesaria de la biblioteca.
* Construir un tablero de Kanban, para visualizar el trabajo y controlar el avance del proyecto.
* Emplear el lenguaje de programación y las herramientas adecuadas, para implementar (programar) el sistema de información.

## **Alcances**

El sistema de información tiene los siguientes alcances:

− Registro de alquiler de canchas

− Realizar reservas.

− Cancelación o anulación de reservas.

− Registro de clientes.

− Búsqueda de canchas disponibles.

− Generación de reportes.

## **1.5 Ingeniería del proyecto**

Para el desarrollo del proyecto se toma en cuenta los tres pilares de la ingeniería de software. (Triángulo de éxito de la ingeniería de software)

1. **Metodología o proceso de desarrollo**
   * Marco de trabajo ágil **SCRUM**, como proceso de desarrollo
2. **Notación de Modelado**
   * Lenguaje de Modelado Unificado (UML)
3. **Herramientas**
   * Framework de desarrollo: .NET Framework
   * Lenguaje de programación: C#
   * Entorno de Desarrollo Integrado (IDE): Visual Studio
   * Manejador de Base de Datos: SQLite
   * Enterprise Architect, para los diagramas UML
   * Balsamiq, para el diseño de interfaces de usuario

# **MARCO TEÓRICO**

//Si se puede incluir figuras

## **2.1 Algoritmo**

Un algoritmo es un conjunto finito de pasos bien definidos y ordenados que permite resolver un problema o realizar una tarea específica. En el contexto del alquiler de canchas de Wally, se pueden utilizar algoritmos para optimizar procesos como la asignación de horarios, la gestión de reservas y la generación de informes.

## **2.2 Software**

El software es un conjunto de programas informáticos, instrucciones y datos que permiten realizar diferentes tareas en un ordenador o dispositivo electrónico. En el contexto del alquiler de canchas de Wally, se pueden utilizar diferentes tipos de software, como sistemas de gestión de reservas, aplicaciones móviles para usuarios y herramientas de análisis de datos.

**2.3 Sistema de información**

Un sistema de información es un conjunto organizado de componentes que interactúan entre sí para recopilar, almacenar, procesar y distribuir información dentro de una organización. En el contexto del alquiler de canchas de Wally, un sistema de información podría incluir una base de datos de clientes y reservas, un sistema de gestión de reservas y una interfaz de usuario para usuarios y administradores.

**2.4 Scrum**

Scrum es un marco de trabajo ágil utilizado en el desarrollo de software que se centra en la entrega iterativa e incremental de productos. En el contexto del desarrollo de un sistema de alquiler de canchas de Wally, Scrum podría utilizarse para gestionar el proceso de desarrollo, priorizar tareas y adaptarse a los cambios en los requisitos del cliente.

**2.5 Modelado**

El modelado se refiere al proceso de representar un sistema o proceso utilizando diferentes tipos de modelos. En el contexto del desarrollo de software para el alquiler de canchas de Wally, el modelado puede incluir la creación de diagramas de flujo, diagramas de clases y otros tipos de diagramas para visualizar y comprender el sistema.

## **2.6 UML: Lenguaje de Modelado Unificado**

UML es un lenguaje estándar utilizado en el modelado de sistemas de software. Incluye diferentes tipos de diagramas que permiten representar diferentes aspectos de un sistema.

### **2.6.1 Diagrama de casos de uso**

Un diagrama de casos de uso en UML representa las interacciones entre un sistema y sus usuarios o actores externos. En el contexto del alquiler de canchas de Wally, un diagrama de casos de uso podría mostrar las diferentes acciones que pueden realizar los usuarios, como realizar una reserva, cancelar una reserva o consultar la disponibilidad de canchas.

### **2.6.2 Diagrama de clases**

Un diagrama de clases en UML representa la estructura estática de un sistema, incluyendo las clases, sus atributos y sus relaciones. En el contexto del alquiler de canchas de Wally, un diagrama de clases podría mostrar las clases relevantes, como Cliente, Reserva y Cancha, y cómo se relacionan entre sí.

### **2.6.3 Diagrama de despliegue**

Un diagrama de despliegue en UML representa la distribución física de los componentes de un sistema en el entorno de ejecución. En el contexto del alquiler de canchas de Wally, un diagrama de despliegue podría mostrar cómo se distribuyen los diferentes componentes del sistema, como la aplicación web y la base de datos, en los servidores de la instalación deportiva.

**2.7 .Net Framework**

.NET Framework es un entorno de desarrollo de software desarrollado por Microsoft que proporciona una amplia variedad de bibliotecas y herramientas para el desarrollo de aplicaciones en diferentes plataformas. En el contexto del desarrollo de software para el alquiler de canchas de Wally, .NET Framework podría utilizarse para desarrollar aplicaciones web, servicios web y otros componentes del sistema.

## **2.8 C#.NET Core**

C# es un lenguaje de programación desarrollado por Microsoft que se utiliza comúnmente en el desarrollo de aplicaciones en el entorno de .NET Framework. .NET Core es una versión multiplataforma y de código abierto de .NET Framework. En el contexto del desarrollo de software para el alquiler de canchas de Wally, C#.NET Core podría utilizarse para desarrollar la lógica de negocio y los componentes de la aplicación.

**2.9 ASP.NET Core**

ASP.NET Core es un marco de trabajo para el desarrollo de aplicaciones web desarrollado por Microsoft y basado en .NET Core. En el contexto del desarrollo de software para el alquiler de canchas de Wally, ASP.NET Core podría utilizarse para desarrollar la interfaz de usuario y la lógica de presentación de la aplicación web.

**2.10 SQLite**

SQLite es una herramienta poderosa y sencilla para gestionar bases de datos en aplicaciones pequeñas a medianas. Su integración directa con la aplicación y su uso de un solo archivo hacen que sea ideal para proyectos que necesitan una base de datos ligera y fácil de gestionar. Con los comandos básicos presentados, puedes empezar a trabajar con SQLite en tus proyectos.

**2.11 Arquitectura MVC**

MVC (Modelo-Vista-Controlador) es un patrón de diseño de software que separa una aplicación en tres componentes principales: el Modelo, que representa los datos y la lógica de negocio; la Vista, que representa la interfaz de usuario; y el Controlador, que actúa como intermediario entre el Modelo y la Vista. En el contexto del desarrollo de software para el alquiler de canchas de Wally, la arquitectura MVC podría utilizarse para organizar y estructurar la aplicación de manera modular y escalable.

## **2.12 Enterprise Architect**

Enterprise Architect es una herramienta de modelado y diseño de software desarrollada por Sparx Systems. Permite a los equipos de desarrollo crear y visualizar modelos de sistemas complejos, incluyendo diagramas UML y documentación relacionada. En el contexto del desarrollo de software para el alquiler de canchas de Wally, Enterprise Architect podría utilizarse para diseñar y documentar la arquitectura del sistema y sus componentes.

## **2.13 Balsamiq**

Balsamiq es una herramienta de diseño de interfaces de usuario que permite a los diseñadores crear prototipos de manera rápida y sencilla. En el contexto del desarrollo de software para el alquiler de canchas de Wally, Balsamiq podría utilizarse para diseñar y prototipar la interfaz de usuario de la aplicación web y móvil, permitiendo a los desarrolladores y diseñadores obtener retroalimentación temprana de los usuarios.

# **MARCO PROPOSITIVO**

## **Diagrama de contexto**

Se identifican los usuarios que van a usar el sistema del Alquiler de Cancha de Wally

//quitar el fondo azul y un poco más profesional pongan a cada actor

Secretaria

Cliente

Propietario

Figura 1: Diagrama de contexto – Alquiler de Cancha de Wally

## **Equipo Scrum y los Stakeholders**

Con fines académicos, los roles de Scrum están representados por:

|  |  |
| --- | --- |
| **Product Owner** | Ronald Mamani Colque |
| **Scrum Master** | Angel Dario Varela Poma |
| **Development Team** | Ronald Mamani Colque  Angel Dario Varela Poma  Wilfredo Villca Flores  David Jose Roque Chaparro |
| **Client** | Dueño de las canchas de wally ( ANDYNA ). |

Tabla 1: Equipo Scrum

## **Historias de Usuario**

Para la determinación de requerimientos del sistema se recolectan historias de usuario.

//completar

|  |
| --- |
| **HU1: Escribir Historias de Usuario** |

Tabla 2: Historia de usuario – Escribir Historias de usuario

|  |
| --- |
| **HU2: Preparar el Entorno de Desarrollo** |

Tabla 3: Historia de usuario – Preparar el Entorno de Desarrollo

|  |
| --- |
| **HU3: Investigar Herramientas** |

Tabla 4: Historia de usuario – Investigar Herramientas

Secretaria:

|  |  |
| --- | --- |
| **HU4: Autenticación al sistema** | |
| **Como** | Secretaria |
| **Quiero** | Ingresar al sistema |
| **Para** | Ver la información y modificar los datos |

Tabla 5: Historia de usuario – Loguin

|  |  |
| --- | --- |
| **HU5: Gestionar Canchas** | |
| **Como** | Secretaria |
| **Quiero** | Llevar un registro de todas las canchas |
| **Para** | Saber si están libres, ocupadas o reservadas |

Tabla 6: Historia de usuario – Registro de Canchas

|  |  |
| --- | --- |
| **HU6: Generar Reporte** | |
| **Como** | Secretaria |
| **Quiero** | Generar un reporte de lo recaudado |
| **Para** | Hacer el seguimiento del mes y año |

Tabla 7: Historia de usuario – Generar Reporte

|  |  |
| --- | --- |
| **HU7: Generar Recibo** | |
| **Como** | Secretaria |
| **Quiero** | Generar un Recibo |
| **Para** | Entrega al cliente como respaldo de pago |

Tabla 8: Historia de usuario – Generar Recibo

Cliente:

|  |  |
| --- | --- |
| **HU8: Ubicación de la Cancha** | |
| **Como** | Cliente |
| **Quiero** | Ver la ubicación de la cancha de wally |
| **Para** | Poder llegar al lugar con facilidad |

Tabla 9: Historia de usuario – Ubicación de la cancha

|  |  |
| --- | --- |
| **HU9: Cancha de Wally Disponible** | |
| **Como** | Cliente |
| **Quiero** | Ver las canchas |
| **Para** | Saber si esta libre, ocupado o reservado. |

Tabla 10: Historia de usuario – Cancha de wally disponible

|  |  |
| --- | --- |
| **HU10: Registrar Flete** | |
| **Como** | Cliente |
| **Quiero** | Registrar mi reserva |
| **Para** | Que lo tengan registrado la cancha |

Tabla 11: Historia de usuario – Registrar flete

Propietario:

No hace login

No hace gestión de secretarias

|  |  |
| --- | --- |
| **HU11: Acceso Total al Sistema** | |
| **Como** | Propietario |
| **Quiero** | Tener acero a todas las opciones del sistema |
| **Para** | Hacer modificaciones que se vean convenientes |

Tabla 12: Historia de usuario – Acceso total al sistema

## **Producto Backlog**

La pila del producto de pendientes a desarrollar está constituida por las historias de usuario (Requerimientos funcionales), y ordenada según prioridad de implementación.

//Corregir tabla y las prioridades máximo será 3 o 4

|  |  |  |
| --- | --- | --- |
| **Historia de**  **usuario** | **Descripción** | **Prioridad** |
| **HU1** | Escribir historias de usuario | 0 |
| **HU2** | Preparar el entorno de desarrollo | 0 |
| **HU3** | Investigar herramientas | 0 |
| **HU4** | Loguin | 1 |
| **HU5** | Registro de canchas | 1 |
| **HU6** | Generar reporte | 2 |
| **HU7** | Generar recibo | 2 |
| **HU8** | Ubicación de la cancha | 2 |
| **HU9** | Cancha de wally disponible | 3 |
| **HU10** | Registrar flete | 3 |
| **HU11** | Acceso total al sistema | 1 |

Tabla 3: Product Backlog

## **Release Planning**

En el plan de despliegue se identifican las iteraciones (**Sprints Backlog**) y los entregables que se van a realizan durante el transcurso del proyecto.

Falta

|  |  |  |
| --- | --- | --- |
| **Sprint / Entregable** | **HU** | **Descripción** |
| **Primer Sprint** | **HU1** |  |
| **HU2** |  |
| **HU10** |  |
| **Segundo Sprint** | **HU3** |  |
| **HU4** |  |
| **HU7** |  |
| **Tercer Sprint** | **HU5** |  |
| **HU6** |  |
| **HU8** |  |
| **HU9** |  |

Tabla 13: Sprint Backlog

## **Diagrama de casos de uso del sistema**

El diagrama de casos de uso del sistema, muestra la funcionalidad global del sistema de información desde el punto de vista de los actores o usuarios.

Corregir con lo observado

No hay gestionar cliente en tus US



Figura 2: Diagrama de casos de uso

## **Diagrama de clases**

El diagrama de clases persistentes muestra las clases que servirán para el diseño de la base de datos



Figura 3: Diagrama de clases

## **Diagrama de Despliegue**



## **Modelo Relacional de la Base de Datos**

A partir del diagrama de clases persistentes diseñamos el modelo relacional de la base de datos

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 4: Modelo Relacional de la Base de Datos

//falta bibliogr