**S.I.G.T.**

Análisis y Diseño de Aplicaciones Web

TechBoss

| **ROL** | **C.I** | **APELLIDO** | **NOMBRE** | **E-MAIL** | **TEL/CEL** |
| --- | --- | --- | --- | --- | --- |
| Coordinador | 5.559.252-7 | Veneri | Martin | martinchico27@gmail.com | 095278843 |
| Sub-coordinador | 5.435.903-9 | Bobadilla | Julieta | julietabobadilla123@gmail.com | 093366757 |
| Integrante 1 | 5.565.388-8 | Mendez | Andrew | andrewnicolas145@gmail.com | 097088873 |
| Integrante 2 | 5.450.638-7 | Fernández | Victoria | Victoriadollanart777@gmail.com | 093354000 |

**Docente: Lippo Galan, Adriano**



**Índice:**

Relevamiento……………………………………..

Elaboración de formularios apropiados para la tarea anterior……………………………………………………………………2

Lógica de sistema……………………………………………………….

Árboles de decisión……………………………………………………..2

Estudio de factibilidades……………………………………………….8

Definición de roles de usuario con permisos y privilegios…………………12

Planificación……………………………………………………………..12

Especificación de Requerimientos (Funcionales Y no Funcionales) ……………………………..……….…….………………….12

Alcance…………………………………………….………………………..13

Limitaciones………………………………………………………………….14

Implementación de metodología Agile en coordinación con el Profesor de gestión de

Proyecto (Propuesta y aplicación)……………………………………………14

Historia de usuarios (Product backlog - Sprint backlog) en coordinación con el

Profesor de gestión de Proyecto (Propuesta y aplicación)...............15

Análisis Costo-Beneficio………………………………….16

Cálculo de métricas del proyecto…………………………… 17

Plan de testing………18

Manual de instalación y mantenimiento de la aplicación…..20

**1**

○ Elaboración de formularios apropiados para la tarea anterior.

Formulario afuera de este pdf

○ Árboles de decisión

2

# Estudio de factibilidades

1. **Factibilidad económica**

Según el cálculo de métricas, SISGRAN estaría generando una ganancia de 8600 dólares estadounidenses y teniendo en cuenta los siguientes costos se calculará el beneficio que conseguiremos al finalizar el proyecto:

* **Costos por única vez:**

| Cantidad | Descripción | Importe (US$) |
| --- | --- | --- |
| 2 | Bloc de hojas A4 | 10 |
| 5 | Escritorios de oficina | 420 |
| 5 | Maquinas para ofimática (todo incluido) | 2395 |
| 5 | Licencias de office OEM | 139 |
| 1 | Impresora Epson Expression XP-2101 | 126 |
|  | Total: | 3090 |

* **Costos recurrentes:**

| Cantidad | Descripción | Importe (US$) |
| --- | --- | --- |
| 1 | Servicio de internet | 43 |
|  | Total: | 43 |

3

* **Beneficios recurrentes:**

| Descripción | Importe (US$) |
| --- | --- |
| Servicio de soporte técnico | 72 |
| Total: | 2 |

* **Comparativo de costos**

| Sol | Costos por única vez | Costos recurrentes mensuales |
| --- | --- | --- |
| 1 | US$ 10 | US$ 43 |
| 2 | US$ 420 |  |
| 3 | US$ 2395 |  |
| 4 | US$ 139 |  |
| 5 | US$ 126 |  |

4

# Factibilidad legal

Se declara que nuestra empresa acepta los acuerdos de usuario proporcionados por cada programa utilizado para la creación del proyecto, las licencias son las siguientes:

* Visual Studio Code: MIT License
* Diagramas Día: GNU General Public License
* Wamp Server: GNU General Public License
* Fedora Linux: GNU General Public License
* VirtualBox: Privativa / GPLv2
* GanttProject: GNU General Public License
* Bitrix24: CRM Free License
* Google Drive: Free plan under terms of service
* Ciertos elementos que utilizamos por el momento no cumplen con la ley pero van a ser regularizados lo antes posible:

5

* Windows 10: Windows 10 professional license

Costo: US$ 193

* Adobe: Creative Cloud License

Costo: US$ 84.99/c. mes

Todos los programas utilizados exceptuando la suite de Adobe, fueron proporcionados por el cliente.

1. **Factibilidad operativa**

Esta parte de la factibilidad comprende una resolución de la probabilidad de que el sistema no sea ambiguo u obsoleto, también para determinar si el sistema es fácil de utilizar por los usuarios.

* Si el sistema es demasiado complejo para los usuarios, es decir, no va a poder alcanzar su cometido o máximo potencial, ya que al uso indebido este podría generar fallas o errores en el sistema.
* Si el sistema introduce cambios muy rápidos el usuario no va a poder adaptarse, es decir, el usuario no va a comprender que está sucediendo con el sistema generando lo que se habla en el punto anterior.
* Si el sistema se puede automatizar solo, esto generaría que el usuario se resista, es decir, que el usuario tenga miedo a ser desplazado (remplazado por el sistema), y así perder su trabajo.
* Otra consideración podría ser que el sistema pueda ser remplazado por la tecnología actualizada de manera que el sistema quede obsoleto, es decir,

6

* que los cambios sean muy rápidos de manera que la tecnología usada actualmente sea remplazada muy fácil haciendo que el sistema quede obsoleto.

1. **Factibilidad técnica**

Nuestro personal dispone de conocimientos, habilidades y equipos, también cuenta con herramientas proporcionadas por el cliente, todas estas cualidades son necesarias para llevar a cabo los procedimientos y funciones del proyecto en sí mismo.

1. **Sistema operativo:**

Este componente es uno de los más importantes ya que tiene que cumplir ciertas características, estas son: Estabilidad; Velocidad; Administración; Fácil uso; Seguridad; Multiusuario y Escalabilidad, ya que necesita soportar la instalación del sistema, brindar velocidad de conexión a las bases de datos y seguridad a los usuarios.

A continuación, se contemplaría diferentes sistemas operativos que cumplan las características mencionadas anteriormente e indispensables para el buen funcionamiento del sistema:

* Windows 10
* Windows 11
* Fedora Server

7

1. **Lenguaje de desarrollo:**

Este elemento debe poseer las siguientes características:

* Facilidad de desarrollo de sistemas
* Fácil de administrar
* Estable y ampliamente usado en ambientes web
* Actualizado constantemente

A continuación, se estaría visualizando lenguajes que cumplan las características mencionadas antes:

* PHP
* Java Script
* CSS
* HTML

1. **Sistema gestor de base de datos:**

Este factor es importante ya que especificara la manera en que se almacenara los datos, la velocidad de procesamiento, el respaldo de datos y la seguridad.

El sistema de gestor de base de datos debe cumplir con los siguientes requisitos:

8

* Estable
* Seguro
* Escalable
* Continuamente actualizado
* Con diferentes lenguajes de programación vía ODBC
* Capaz de almacenar inmensas cantidades de información

A continuación, se presentan gestores de bases de datos que cumplen con los requisitos mencionados:

* MySQL
* SQL Server
* MariaDB
* PostgreSQL
* mongoDB

1. **Características del Hardware disponible para el desarrollo:**

|  | Elemento | Capacidad |
| --- | --- | --- |
| Equipo 1 | Memoria RAM | DDR3 8 GB |
| Disco SSD | 240 GB |
| Procesador | Intel® Core™ i7 4790 |
| CD-ROM/DVD-ROM | Si |
| Equipo 2 | Memoria RAM | DDR3 8 GB |
| Disco SSD | 240 GB |
| Procesador | Intel® Core™ i7 4790 |
| CD-ROM/DVD-ROM | Si |
| Equipo 3 | Memoria RAM | DDR3 8 GB |
|  |  |
| Disco SSD | 240 GB |
| Procesador | Intel® Core™ i7 4790 |
| CD-ROM/DVD-ROM | Si |
| Equipo 4 | Memoria RAM | DDR3 8 GB |
| Disco SSD | 240 GB |
| Procesador | Intel® Core™ i7 4790 |
| CD-ROM/DVD-ROM | Si |
| Equipo 5 | Memoria RAM | DDR3 8 GB |
| Disco SSD | 240 GB |
| Procesador | Intel® Core™ i7 4790 |
| CD-ROM/DVD-ROM | Si |
| Servidor  PC-1 | Memoria RAM | DDR3 8 GB |
| Disco SSD | 240 GB |
| Procesador | Intel® Core™ i7 4790 |
| CD-ROM/DVD-ROM | Si |

Con lo anterior presentado podemos decir que el equipo de desarrollo es lo suficientemente robusto y maduros para desarrollar y soportar el sistema, además son proporcionadas por el cliente para desarrollar el sistema.

1. **Experiencia y conocimiento del equipo de desarrollo:**

Los Recursos Humanos, experiencia y conocimiento personal se exponen a continuación:

10

| Recursos Humanos | Coordinador |
| --- | --- |
| Sub-coordinador |
| Dos integrantes |
| Experiencia | Administración de proyectos |
| Desarrollo de sistemas |
| Trabajo en equipo |
| Programación |
| Base de datos |
| Diseño |
| Conocimiento | Lenguaje de Programación Web |
| UML |
| Base de Datos Relacionales |

Anteriormente se detalla el recurso humano del que se dispone para el desarrollo del sistema: El Coordinador como guía principal del proyecto y encargado de la verificación, y validación del trabajo realizado;

El Sub-Coordinador encargado de ayudar al Coordinador en caso de exceso de trabajo, también está encargado de la supervisión en caso de que el Coordinador no se encuentre en el momento; los integrantes están distribuidos y encargados en diferentes áreas las cuales se les da mejor trabajar, estos cada que terminan una tarea la entregan al Coordinador para una verificación, en caso de que no esté correcto, le dará consejos para poder arreglar dicho error.

Por lo anterior dicho, podemos decir que el personal está sobre calificado técnicamente, ya que posee el conocimiento y la capacidad necesaria para cumplir con los requisitos y concluir con éxito dicho sistema.

11

**● Definición de roles de usuario con permisos y privilegios.**

Administradora: Julieta Bobadilla.

Sera capaz de administrar sin restricción alguna el sistema.

Supervisores: Victoria Fernández, Martin Veneri.

Estarán pendientes al bienestar del sistema como por ejemplo el almacenamiento, sesiones y su información, conexiones y más sin embargo deben de recurrir al administrador debido a que no pueden realizar ningún cambio.

Mantenimiento del sistema: Andrew Méndez.

Estará constantemente haciendo actualizaciones para que el sistema no quede desactualizado, mejorando el rendimiento del sistema y agregando nuevas funcionalidades de ser necesario.

**● Planificación.**

Para la planificación hicimos un gantt para tener todo bien detallado.

Lo primero en planificar son los tiempos que hay entre una entrega y otra ya que para llevar a cabo este proyecto nos debemos juntar y no todos siempre tenemos los mismos horarios disponibles. Después los requerimientos de cada tarea ya que eso es lo más importante en esta planificación.

**● Especificación de Requerimientos**

Los  requerimientos funcionales son esenciales para el desarrollo del software, ya que proporcionan una guía clara sobre las características que deben ser implementadas y sirven como base para realizar pruebas y verificar si el software cumple con las necesidades del usuario y  los requerimientos no funcionales,  se centran en aspectos

como el rendimiento, la seguridad y control de acceso, la interfaz con otros sistemas y otros atributos del sistema que complementan los requerimientos funcionales.

12

**Alcance**

El alcance de este proyecto es renovar la página web de la confederación uruguaya de karate, también la creación de una aplicación móvil con funciones específicas para mejorar su base de datos.

**Limitaciones**

Una de las principales limitaciones es que todos los integrantes de la empresa tengan las mismas versiones de las aplicaciones que se necesitan utilizar para llevar a cabo este proyecto.

El software debe ser seguro  para proteger la información y los sistemas contra amenazas y ataques cibernéticos

El software debe ser adaptable al crecimiento de usuarios y a las funcionalidades futuras que necesite en su momento. Si no tenemos en cuenta esta limitación desde el principio, puede ser difícil adaptar el software a medida que aumentan los requerimientos o el número de usuarios.

13

**● Implementación de metodología Agile en coordinación con el Profesor de gestión de**

**Proyecto (Propuesta y aplicación).**

**Historia de usuarios:**

Product Backlog: Crear tipos de usuarios (juez): Como sistema, quiero permitir la creación de dos tipos de usuarios: estándar y administrador, para poder gestionar adecuadamente las funciones y permisos en el sistema.

Iniciar sesión (juez): Como usuario juez, quiero poder iniciar sesión en el sistema proporcionando un nombre de usuario y contraseña válidos para acceder a mis funciones.

Puntuar participante (juez): Como juez, quiero tener la capacidad de puntuar a los participantes desde una lista para evaluar su desempeño en el evento.

Avanzar al siguiente participante (juez): Como juez, quiero tener la opción de avanzar al siguiente participante después de puntuar para clasificar a los participantes de manera eficiente.

Ingresar al sistema como técnico (técnico): Como técnico, quiero poder acceder al sistema en caso de cualquier problema o necesidad de mantenimiento.

Sprint Backlog (para un Sprint específico): Durante el Sprint X, se priorizarán y seleccionarán las siguientes tareas del Product Backlog.

Iniciar sesión (juez): Implementar el formulario de inicio de sesión que requiere usuario y contraseña para el usuario juez.

Crear tipos de usuarios (juez): Implementar la funcionalidad que permite la creación de dos tipos de usuarios: estándar y administrador.

Puntuar participante (juez): Desarrollar la capacidad de puntuar a los participantes desde una lista y registrar esas puntuaciones en el sistema.

Avanzar al siguiente participante (juez): Implementar la funcionalidad que permite a un juez avanzar al siguiente participante después de puntuar.

Ingresar al sistema como técnico (técnico): Implementar la opción para que un técnico pueda acceder al sistema en caso de problemas o necesidades de mantenimiento. Este Sprint Backlog representa las tareas específicas que se abordarán durante un Sprint determinado. Las tareas se seleccionan en función de la prioridad y la capacidad del equipo para completarlas durante ese período. Después de completar estas tareas, se puede realizar una revisión para demostrar las nuevas funcionalidades implementadas y planificar el siguiente Sprint.

**Análisis costo-beneficio:**

1. Determinación y Cuantificación de Costos**:** Se consideran los costos asociados con el diseño, implementación**,** capacitación**,** mantenimiento y mejora del software.

2. Identificar y cuantificar beneficios**:** Se consideran factores como aumento de la productividad, reducción de costos operativos, aumento de ingresos y satisfacción del usuario.

3. Calcular el índice de fortalezas y debilidades**.** Esto se determina dividiendo tus ingresos totales por tu**s** gastos totales, lo que te da un índice de fortalezas y debilidades**.** Un valor mayor que 1 significa que los ingresos exceden los gastos, lo que significa que el proyecto es rentable**.**

4. Realizar estudios de sensibilidad: Analizar cómo los cambios en factores clave como costos y beneficios afectan la rentabilidad del proyecto.

5. Tome decisiones informadas: aplique análisis de compensaciones y resultados de viabilidad para determinar si un proyecto es apropiado o si se necesitan cambios en el enfoque, los recursos o el cronograma**.**

14

**Cálculo de métricas del Proyecto:**

Nuestro objetivo será proporcionar un software que sea un sistema de puntuación para facilitar la valoración de los jueces sobre los contrincantes.

La meta principal a la que apuntamos es a que nuestro software sea utilizado internacionalmente y no solo en nuestro país.

**Métricas clave:**

Como primera métrica apuntamos a que en tan solo medio año logremos que nuestro software sea utilizado ya en competencias de alto nivel, como las que organiza la CUK (organización uruguaya de karate).

Como segunda apuntamos a que nuestro software alcance los grandes juegos olímpicos. Esto sería un gran mérito.

Las métricas serán calculadas y medidas semanalmente mediante google docs y sus allegados

En en caso de que se de algún imprevisto se de y se retrase, se preverán una semana extra para solucionar este supuesto imprevisto.

Para la recolección de datos lo que vamos a utilizar serán bases de datos, en las cuales se almacenarán los datos obtenidos mediante formularios y encuestas en línea.

Estos datos previos se almacenarán en bases de datos codificadas y monitoreadas por personal técnico en caso de alguna emergencia ocurra.

Previamente a utilizar los datos antes obtenidos debemos de aplicar determinados “filtros” para eliminar duplicados, etc.

Después de filtrar los datos debemos de analizar estos mismos para corroborar semejanzas entre ellos y así averiguar lo que la gente quiere.

Evaluaremos las respuestas anteriores y veremos los puntos en los cuales debemos de mejorar.

Implementar áreas de mejora para perfeccionar nuestros puntos débiles. Luego agregaremos medidas correctivas para así facilitar las cosas en el caso de que en algún futuro se presente algún problema similar.

Estableceremos un sistema de desarrollo continuo para implementar cambios en tiempo real.

Llegados a este punto haremos una revisión desde cero de todo el proyecto para así identificar en las áreas en las cuales estuvimos más flojos para en un futuro no cometer los mismos errores.

Después de todo lo antes mencionado cerraremos el proyecto formalmente para así culminar con todo lo restante.

**Plan de testing**

**Prueba de registro de participante:**

-Verificar que los participantes puedan registrarse correctamente en el sistema.

**Prueba de ingreso de puntuación:**

-Asegurarse de que el sistema permita a los jueces ingresar las puntuaciones de los participantes.

**Prueba de integración Frontend-Backend:**

-Comprobar que el frontend y el backend se integren adecuadamente y se comuniquen correctamente.

**Prueba de visualización de puntuaciones:**

-Verificar que las puntuaciones se muestren correctamente en la interfaz de usuario.

**Prueba de gestión de participantes:**

-confirmar que el sistema pueda gestionar la información de los participantes, incluyendo nombres, apellidos, país de origen y detalles del kata.

**Tabla de decisión para tipos de colores:**

-Verificar que las puntuaciones se asignen correctamente según el color del participante (rojo o azul).

| **Color** | **Puntuación esperada** |
| --- | --- |
| Rojo | Puntuación X |
| Azul | Puntuación Y |

**Tabla de decisión para tipos de kata:**

-Asegurarse de que las puntuaciones se gestionen correctamente según el tipo de kata.

| **Tipo de kata** | **Puntuación esperada** |
| --- | --- |
| Kata A | Puntuación A |
| Kata B | Puntuación B |

**1.Caso de uso: Registrar nuevo participante:**

Pre condiciones: Usuario identificado.

Pasos:

1.1- Navegar a la sección de refriego de participantes.

1.2- Ingresar los datos del nuevo participante

1.3- Guardar la información del participante

**Resultados esperados:**

El participante se registra correctamente al sistema.

**2.Caso de uso: Ingresar puntuación:**

Pre condiociones: Juez autenticado, participante registrado

Pasos:

2.1- Seleccionar al participante y el tipo de kata.

2.2- Ingresar la puntuación asignada por los jueces.

2.3- Guardar la puntuación.

**Resultados esperados:**

La puntuación se registra correctamente para el participante y el tipo de kata seleccionado.

**Caso de uso: Visualizar puntuaciones:**

Pre condiciones: Puntuación Registradas en el sistema.

Pasos:

3.1- Navegar a la selección de visualización de puntuaciones.

3.2- Verificar que se muestren correctamente todas las puntuaciones registradas.

**Resultados esperados:**

Las puntuaciones se muestran correctamente en la interfaz de usuario.

**Notas finales:**

* Realiza pruebas de aceptacion con usuarios reales para obtener retroalimentación adicional.
* Documenta cualquier encontrado y trabaja en su resolución.
* Repite las pruebas después de cada actualización o cambio importante en la aplicación.

**Manual de instalación y mantenimiento de la aplicación**

**Requisitos previos-**

**Entorno de desarrollo:**

Preveer un entorno de desarrollo configurado con las herramientas necesarias, como un editor de código y un navegador web compatible con este.

**Lenguaje de programación:**

Seleccionar un lenguaje de programación que sea adaptable a las necesidades.

**Base de datos:**

Seleccionar que sistema de gestión de bases de datos se utilizara.

**Instalar dependencias:**

Instalar dependencias necesarias para frontend y backend.

**Configurar base de datos:**

Configurar base de datos y sus migraciones si estas son necesarias.

**Pasos de instalacion-**

**Configurar variables de entorno:**

Configurar las respectivas variables de entorno necesarias (claves api, base de datos, etc) .

**Iniciar backend:**

Iniciar el servidor backend.

**Iniciar frontend:**

Iniciar el servidor frontend.

**Acceder a la aplicacion:**

Seleccionar un navegador web y acceder a la aplicación.

**Mantenimiento-**

**Actualizaciones:**

Realizar actualizaciones periodicas de la aplicaciones para corregir errores y agregar nuevas funciones según sea necesario.

**Respaldo de datos:**

Implementa un sistema de respaldo regular para asegurar los datos criticos de la aplicacion.

**Monitoreo y registro:**

Implementar herramientas de monitoreo y registro para rastrear el rendimiento y detectar posibles problemas.

**Seguridad:**

Manten actualizadas las bibliotecas y dependencias para abordar posibles vulnerabilidades de seguridad.