

DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE PARA EL SISTEMA CON NOMBRE EN CLAVE “Aris<T>a”

Versión 0.1

21 de marzo del 2025

Preparado por: El equipo Aris<T>a

Realizó: El equipo Aris<T>a.

CONTROL DE DOCUMENTACIÓN

Control de Configuración

Título:	Plantilla para la Especificación de Requerimientos de Software
Referencia:	
Autor:	Equipo Aris<T>a
Fecha:	21 de marzo del 2025

Histórico de versiones

Versión	Fecha	Estado	Responsable	Nombre de archivo
0.1	21/Mar/2025	B	Equipo Aris<T>a	Levantamiento de Requerimientos.doc

Estado: (B)orrador, (R)evisión, (A)probado

Propósito:

El propósito del presente documento es proporcionar un marco de referencia formal de los deseos, expectativas y necesidades que presenta actualmente nuestro cliente, y de las cuales mediante este proyecto nos encargaremos de satisfacer cada una de ellas. Igualmente, dicho documento igualmente fungirá como documento legal el cual deberá ser debidamente anexado en el contrato de prestación de servicios correspondiente.

Audiencia:

Este documento está orientado a satisfacer las dudas e inquietudes de otros analistas de negocio, así como de otros posibles diseñadores. El lenguaje empleado en este documento es accesible tanto

para personal técnico como no técnico, garantizando una claridad y visibilidad de las acciones que se llevaran a cabo.

Producto a elaborar:

A extremadamente grandes rasgos nuestra aplicación (Arista) consistirá en una agenda automatizada que permitirá gestionar automáticamente las reuniones entre expertos y estudiantes (asemejándose a las actividades que realiza una secretaria en la vida real). La aplicación no será una red social orientada en publicitar eventos comunitarios, tampoco será una plataforma para ofrecer ofertas de trabajo/internships y menos una plataforma de conferencias.

Fecha deseada de inicio y finalización del desarrollo:

La fecha tentativa de inicio del presente proyecto corresponde a 22/03/2025 en la cual se tiene planeado realizar la reunión de “kickoff”, tentativamente, según el calendario anexo en el contrato, el producto deberá ser entregado a más tardar la fecha estipulada en el contrato.

Definiciones, acrónimos y abreviaturas:

Estudiante: Persona actualmente inscrita en la facultad de matemáticas, perteneciente a la Universidad Autónoma de Yucatán, ubicada en Mérida, Yucatán, México.

UADY: Universidad Autónoma de Yucatán

Reclutador: Aquel empleado perteneciente a una empresa privada, el cual desempeña el cargo de “Empleado de Recursos Humanos”, el cual su rol es encontrar aquellos candidatos que satisfagan las vacantes de una empresa.

Empleado de empresa tecnológica: Se refiere a aquel individuo que labora en una empresa privada, el cual ostenta un cargo distinguido y posee cierto grado y años de experiencia que respalden su opinión y conocimientos.

Emprendedor fundador: Nos referimos a aquel individuo el cual por sus propios medios e ingenio empleando sus conocimientos técnicos, así como habilidades blandas, constituyó la creación de un emprendimiento, el cual naturalmente se transformó a una empresa legalmente registrada.

API: “Application Program Interface”

Docente: Cuanto mencionamos a docente nos referimos a aquel trabajador que labora actualmente en la Universidad Autónoma de Yucatán, el cual puede ostentar el grado académico de Licenciado, Ingeniero, Maestro o doctor.

Experto en el área: Cuando empleemos este término a lo largo de estos documentos haremos referencia indistintamente a Reclutadores, Empleados de empresas tecnológicas, Emprendedores fundadores, así como a Docentes. Se les otorga dicha distinción, gracias a su trayectoria, así como a su amplio conocimiento en el área.

Panorama general:

A lo largo del presente documento nos daremos a la tarea de desglosar los deseos, expectativas, anhelos y necesidades de nuestro cliente, con el fin de obtener el mejor entendimiento del dominio del

problema. Con la intención de que este documento sea usado como línea base para los demás procesos del ciclo de desarrollo de software.

Descripción General

Perspectiva del producto:

Como se comentó brevemente al inicio, el producto a generar no constituye como tal una innovación tecnológica, simplemente es un producto de software “inspirado” y/o basado en otros productos de software “populares” en el mercado. Sin embargo, dentro de la justificación del presente documento se exponen las razones por las cuales nuestros clientes desean elaborar dicho producto de software, debido a que sus deseos y necesidades son particulares.

El producto por elaborar es independiente y autocontenido, no representa una nueva versión de algún sistema existente. Tampoco va a sustituir ninguna clase de sistema existente.

El proyecto está constituido por un gran componente, no existen como tales otros subsistemas que formen parte de dicho, y tampoco es parte de algún sistema más grande. Sin embargo presenta principalmente 2 dependencias, la primera respectiva a la base de datos a emplear y la segunda respecto a la API de reuniones virtuales que se va a emplear; este punto se abordara en su respectivo apartado.

El cliente desea (y sobre todo exige) que la solución tecnológica empleada se utilizable/visualizable desde cualquier dispositivo móvil, ya sea Laptop, Teléfono Celular, Tableta; igualmente debe poder apreciarse en equipos informáticos de escritorio. Debe tomarse en cuenta que para todo caso es necesario poder realizar la interacción tanto de manera táctil, como por medio de dispositivos periféricos, principalmente el “mouse”.

De antemano debido a las características de uso que exige el cliente nos encontramos de cierta manera restringidos a soluciones tecnológicas multiplataforma o basadas en web. El cliente expreso claramente que no desea múltiples proyectos para las múltiples plataformas. (Esto debido a que se le comento que, dadas ciertas herramientas tecnológicas, es posible evitar realizar proyectos adicionales.

Interfases del sistema:

Como se mencionó con anterioridad el sistema es autónomo e independiente, no presenta interdependencias o interconexiones con otros sistemas de software actuales o futuros.

Interfases de usuario:

Como se mencionó con anterioridad el sistema debe ser posible de usar en cualquier dispositivo móvil o estático, esto incluye su uso tanto de manera táctil como háptica (usando algún dispositivo como ratón y teclado).

Para proveer de más claridad en la especificación de los requisitos procedemos a exponer los prototipos gráficos (obviamente aun no poseen funcionalidad). Esto con la intención de dejar en claro el resultado final que desea el cliente y por supuesto proveer de certeza al equipo de diseño.

Interfases de hardware:

Debido a la naturaleza del contexto donde se desenvolverá en el software, formalmente no se configurarán explícitamente puertos, máquinas virtuales o cual sea otro tipo de administración de tecnologías de la información. Debido al nulo presupuesto del cliente la solución informática deberá hospedarse en alguna plataforma que ofrezca una capa de servicio gratuito. La plataforma específica, así como la configuración concreta del sistema dependerá de la solución tecnológica provista en el apartado de construcción.

Formalmente, aunque el software se alojara en servidores auto hospedados es necesario que todo dispositivo comercial de clase civil (Teléfonos, Tablet, Laptops, etc.) pueda acceder a la solución tecnológica sin mayor complicación o configuración.

Interfases de software:

Como se mencionó brevemente con anterioridad dadas las necesidades y deseos del cliente se identificaron 3 dependencias principales para la elaboración del software. Aunque este documento no abordara el cómo se deberán emplear estas interfaces, es necesario tomarlas en consideración al momento del diseño ya que los métodos/formas los cuales operan estos “paquetes” de software resultan volátiles, esto igualmente deberá considerarse en el plan de administración de riesgos. Y aunque existen otras dependencias de software, dichas son menores y en gran medida constituyen necesidades propias del área de construcción de software, por lo que por el momento las consideraremos despreciables. A continuación, se enlistarán las dependencias “mayores”:

Dependencia 1:

Nombre formal: manejador de base de datos MongoDB

Mnemónico: MongoDB

Número de especificación: ERS-InterazS-A1

Número de versión: 8.0.0 (última versión estable)

Fuente: Puede consultarse los artefactos de software en [Download MongoDB Community Server | MongoDB](#)

Interacción: Como base de datos no relacional será la encargada de almacenar todos los datos generados por la aplicación.

Dependencia 2:

Nombre formal: Google OAuth2 Public Api

Mnemónico: Autenticación

Número de especificación: ERS-InterazS-A2

Número de versión: 3.2.5

Fuente: Puede consultarse los artefactos de software mediante el software como servicio que brinda “Firebase” (Propiedad de Google)

Interacción: Como sistema de autenticación basado en inicio de sesión de Google provee una autenticación rápida y sin necesidad de contraseñas.

Dependencia 3:

Nombre formal: Google Calendar Public API

Mnemónico: Reuniones virtuales

Número de especificación: ERS-InterazS-A3

Número de versión: 2.0.1

Fuente: Puede consultarse los artefactos de software mediante el Marketplace correspondiente a la plataforma de Google Cloud.

Interacción: El sistema permite la generación de una o más reuniones virtuales las cuales implican la participación de 1 o más personas, mediante la plataforma Google Meet.

Aclaración: Por alguna extraña razón las funcionalidades de generación de reuniones pertenecen al API de Google Calendar, se realiza esta aclaración explícita con la finalidad de evitar confusiones al momento de realizar el diseño del sistema.

Interfases de comunicación:

Debido a la naturaleza de la aplicación en la cual al cliente no le interesa emplear la última tecnología, ni tampoco presenta interdependencia con otras interfaces de sistemas. Se sugiere al equipo usar el clásico protocolo http.

El argumento de esta elección es debido a que en la facultad de matemáticas durante la asignatura impartida durante el quinto semestre denominada “Desarrollo de aplicaciones web”, los estudiantes reciben instrucción en dicho tema. Lo que puede favorecer que en algún momento los alumnos puedan proporcionar mantenimiento al sistema informático, ya que comprenden las bases fundamentales de cómo opera el sistema.

Restricciones de memoria:

Como se mencionó desde un inicio los fondos monetarios de nuestro cliente son nulos. Eso implica que la plataforma donde se va a alojar nuestra solución tecnológica se verá limitada por la capa gratuita del servicio. Después de consultar múltiples aplicaciones de alojamiento de sistemas informáticas, el equipo de analistas de negocio, llegó a la conclusión que el promedio de memoria RAM para esta plataforma se encuentra limitada a 512 Mb.

Requerimientos de adaptación del sitio:

A lo largo del presente documento se ha realizado énfasis que el producto informático a entregar (sistema) no estará empotrado, aislado o controlado. La intención es que se pueda usar en cualquier espacio, siempre y cuando se posea una conexión estable a internet. Cabe aclarar que durante el refinamiento de la interfaz gráfica deben considerarse aspectos como múltiples tipos

de iluminación, tiempos de espera debido a pésima conexión, así como cualquier otro aspecto que pueda arruinar la experiencia del usuario.

Funciones del producto:

Al inicio de este documento nos dimos a la tarea de tratar de ejemplificar de la manera más simple a grandes rasgos que es el producto. En su momento mencionamos que era una agenda automatizada. Dicha definición está muy lejos de ser lo que realmente desea el cliente, sin embargo, es un buen punto de partida para tratar de concretar el resultado que desea y sobre todo espera el cliente.

Al menos en esta sección nos abstendremos a mencionar funcionalidades las cuales no sean vitales (desde la perspectiva del cliente) con la finalidad de contextualizar el resultado deseado.

- Un experto en el área (revisar definiciones y acrónimos) puede ofrecer una fecha y horario de asesoría.
- Existe una cantidad finita de expertos en el área.
- Un estudiante puede visualizar la n cantidad de reuniones que ofrecen los n expertos en el área registrados en el sistema.
- Cada experto en el área posee un calendario digital (el cual se asemeja a un calendario físico); en dicho calendario digital se menciona de forma estructura las reuniones que ofrece el experto en el área.
- Un estudiante puede observar las reuniones específicas que ofrece el experto en el área X.
- Un estudiante se puede postular a una o más reuniones ofrecidas por X experto en el área.
- Múltiples estudiantes se pueden postular a una reunión específica.
- Cada experto en el área, dadas las reuniones que ofreció, puede seleccionar al o los candidatos con quien desee llevar a cabo la reunión.
- Una vez seleccionados los candidatos seleccionados (por reunión ofrecida) se notifican a las partes correspondientes.

Restricciones generales:

Nuestro sistema al ser un sistema oficial perteneciente a la UADY debe cumplir ciertas normativas específicas de la institución con el fin de garantizar el reconocimiento de “oficialidad” de nuestro sistema de software. A continuación, mencionaremos dichas restricciones que deben tomarse en cuenta durante todo el proceso de desarrollo.

Políticas regulatorias:

- Deben emplearse los colores institucionales, los cuales son:.....
- El logo de la UADY debe presentarse en todo momento en la esquina superior izquierda de la página.
- Toda página debe de contar con el pie de página donde se mencionen las redes sociales, así como el aviso de privacidad correspondiente.
- Cuando se presente alguna fotografía/imagen referente al jaguar, dicha foto no debería sobajar/ridiculizar al jaguar; debido a que es un símbolo institucional.
- Cada página debe presentar la leyenda “© Todos los Derechos Reservados, UADY 2025”
- Cada página debe presentar la leyenda "Luz, Ciencia y Verdad"

Limitaciones de hardware:

Como se mencionó un principio debido al nulo presupuesto que presenta el proyecto, nos veremos en la necesidad de emplear instancias gratuitas de plataformas de alojamiento de software; aunque aún no se ha decidido que plataforma se va a emplear, es necesario tomar en consideración las limitaciones existentes para el hardware en su capa gratuita, a manera de ejemplo se anexa una referencia de un servicio líder en el mercado. Find a plan to power your projects.

Funciones de auditoría:

Debido a que la plataforma va a ser empleada tanto por directivos, coordinadores de carreras, secretarios académicos y administrativos (dichos corresponden a nuestros usuarios terciarios), es necesario mantener una monitoriabilidad respecto a todas las reuniones que se lleven a cabo en la plataforma con la finalidad de tomar las mejores decisiones basadas en datos.

Lo crítico de la aplicación:

En el sistema informático existen 2 preocupaciones generales que se deben ser consideradas, la primera es la protección de los datos personales, no deben de quedar expuestas por ningún medio contraseñas, correos electrónicos o cualquier otro dato que invada la privacidad de nuestros usuarios, de igual manera otro aspecto crítico del producto es la precisión de la generación de las reuniones, se deben de considerar la precisión de las zonas horarias así como los cambios de horario para garantizar la máxima exactitud al momento de celebrar las reuniones correspondientes.

Supuestos y dependencias:

Como se mencionó en un punto anterior existe una fuerte dependencia con los sistemas mencionados en el apartado “interfaces de software”. Debido a que cambian constantemente (y sin contar que la documentación es ambigua) representa un gran factor de riesgo para el éxito del proyecto por lo que todo diseño e implementación deben ser monitoreados con cautela.

Asignación de requerimientos:

Reiterando lo mencionado durante todo este documento el proyecto, así como todas sus funcionalidades representan la primera versión. Cabe aclarar que el desempeño del sistema, así como el grado en que satisfaga las necesidades de los clientes, será un indicador clave para determinar la continuidad del proyecto, por lo que se espera que los requerimientos enumerados en las siguientes secciones sean los suficientes para “enamorar” a nuestros clientes para que este proyecto pueda evolucionar, y sobre todo madurar.

Requerimientos específicos

Interfaces externas.

Como se mencionó en el apartado inmediato anterior sería completamente inútil tratar de describir las entradas y salidas de las interfaces (en especial las de código) debido a su extremadamente alta

volatilidad, por lo que no se puede garantizar una versión estable para la interfaz (cada 3 meses modifican algo, lo que obliga a realizar mantenimiento correctivo).

Funciones (requerimientos funcionales).

Para mayor facilidad procederemos a separar los requisitos funcionales por “vistas” con tal de lograr una coherencia completa:

Ventana de Inicio de sesión:

Como su nombre lo indica claramente, es la vista orientada a proveer al usuario un método simple de verificación y autenticación.

a) Validación de entradas

Dado que el inicio de sesión se realiza mediante Google (así lo exigió el cliente), no se requiere una validación manual de campos por parte del usuario. La validación de entradas se delega al sistema de autenticación de Google, que incluye:

- Verificar las credenciales del usuario (correo y contraseña).
- Asegurar que el correo esté asociado a una cuenta válida.
- Confirmar que el usuario haya otorgado los permisos necesarios para la autenticación.

b) Secuencias de operación exactas

1. Pantalla de inicio: El usuario visualiza la opción "Iniciar sesión con Google".
2. Clic en el botón de Google: El usuario hace clic en el botón "Iniciar sesión con Google".
3. Redirección al sistema de Google: El usuario es redirigido a la página de inicio de sesión de Google.
4. Autenticación en Google: El usuario ingresa sus credenciales (correo y contraseña) o selecciona una cuenta previamente autenticada.
5. Permisos: El usuario concede permisos al sistema para acceder a su información de perfil (si no lo ha hecho antes).
6. Redirección al sistema principal: Una vez autenticado, el usuario es redirigido nuevamente a la aplicación principal, donde se le otorga acceso.

c) Respuestas a situaciones anormales

1. Sobreflujo:

Escenario: Si múltiples usuarios intentan iniciar sesión simultáneamente.

Respuesta esperada: El sistema debe manejar múltiples solicitudes concurrentes sin afectar el rendimiento. Esto se logra mediante el backend de autenticación de Google, que está diseñado para

alta concurrencia.

2. Facilidades de comunicación:

En caso de fallos, se deben proporcionar mensajes claros al usuario, como:

"No se pudo conectar con Google. Intente nuevamente."

"La autenticación falló. Verifique su conexión a internet."

Estos mensajes deben ser visibles en la interfaz gráfica.

3. Manipulación y recuperación de errores:

Error de red: Mostrar un mensaje como "No se pudo establecer conexión. Verifique su internet e intente de nuevo."

Credenciales incorrectas: Google maneja este error mostrando un mensaje al usuario directamente en su interfaz.

Cuenta no autorizada: Mostrar un mensaje en la aplicación principal indicando: "No se pudo autenticar. Asegúrese de usar una cuenta válida."

d) Efectos de parámetros

Selección de cuenta de Google: Si el usuario tiene múltiples cuentas, el sistema le permite elegir cuál usar. Esto afecta la sesión y los permisos otorgados.

Permisos de Google: Si el usuario no otorga permisos requeridos, la autenticación falla, y el sistema debe redirigirlo nuevamente a la pantalla inicial.

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias:

Entrada: Clic en el botón "Iniciar sesión con Google".

Salida: Redirección a la pantalla de autenticación de Google.

Entrada: Credenciales del usuario (correo y contraseña) ingresadas en Google.

Salida: Token de autenticación que se devuelve al sistema principal para verificar y otorgar acceso.

2. Fórmulas de conversión de entrada/salida:

Las credenciales del usuario no son manejadas directamente por la aplicación principal. Google genera un "token" el cual contiene la meta data del usuario, fecha de generación, entre otros aspectos que no son de interés para el diseño.

Landing page:

Una vez que el usuario inicie sesión exitosamente será redirigido a estas vistas, en la cual encontrara un pequeño espacio de noticias, cabe aclarar que debido a las limitaciones presupuestarias el contenido de las noticias deberá ser estático, es decir, será "hardcodeado" directamente en la interfaz gráfica. A continuación, se mencionarán detalles más formales.

a) Validación de entradas

Dado que las noticias se agregan de manera estática, no hay un sistema dinámico que implique entradas del usuario en esta interfaz. Sin embargo, pueden considerarse las siguientes validaciones:

1. Validación manual al ingresar datos en el código fuente o sistema de gestión:
 - a. Verificar que el título no esté vacío.
 - b. Comprobar que la fecha tenga un formato correcto (por ejemplo, YYYY-MM-DD).
 - c. Validar que el texto de resumen no exceda un límite razonable de caracteres (por ejemplo, 200 caracteres).
 - d. Verificar que el enlace "Ver más" apunte a una URL válida.

b) Secuencias de operación exactas

1. El usuario accede a la sección "Últimas Noticias" (Página inicial por defecto).
2. Visualiza una lista de noticias con:
 - a. Título.
 - b. Fecha.
 - c. Resumen.
 - d. Opción de "Ver más".
3. Si el usuario hace clic en "Ver más", es redirigido a una página o sección estática con más detalles sobre la noticia (esto dependerá de la implementación).

c) Respuestas a situaciones anormales

1. Sobreflujo:

- Escenario: Si se agregan muchas noticias en el diseño estático.
- Respuesta esperada:
 - Limitar la cantidad de noticias visibles por página.
 - Implementar paginación o un sistema de scroll infinito para manejar grandes volúmenes de contenido.

2. Facilidades de comunicación:

- En caso de que no haya noticias disponibles, mostrar un mensaje claro como:
 - "No hay noticias disponibles en este momento."

3. Manipulación y recuperación de errores:

- Error de enlace roto en "Ver más": Desgraciadamente este apartado debe validarse manualmente.

d) Efectos de parámetros

Los parámetros que afectan la visualización de noticias incluyen:

1. Cantidad de noticias:
 - a. Debe evitarse sobre saturar la página de noticias, lo que puede provocar una sobreestimulación, lo cual conlleva a una pobre experiencia de usuario.
2. Longitud del texto:
 - a. Debe cuidarse que el texto no sea demasiado largo, ya que puede dar lugar a desbordamientos poco agradables a la vista.

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias:

- Entrada: Datos estáticos proporcionados en el sistema o código fuente, incluyendo:
 - Títulos de noticias.
 - Fechas de publicación.
 - Resúmenes.
 - URLs para "Ver más".
- Salida: Interfaz que muestra las noticias en el orden definido con los elementos visuales correspondientes.

2. Fórmulas de conversión de entrada/salida:

- No aplican

Ver mi perfil

En esta sección/apartado se mostrarán los datos que componen tu perfil correspondiente a la página, el fin de dicho es exhibir de manera clara, concisa y visualmente agradable las características de cada individuo.

Para evitar perder la continuidad aprovecharemos la oportunidad de mencionar la interfaz gráfica denominada "Editar perfil", la cual como su nombre lo indica se encarga de modificar los datos personales de cada usuario, a continuación, se mencionarán con mayor detalle:

1. Vista de perfil: Muestra los datos personales del usuario y botones para editar el perfil o cerrar sesión.

2. Vista de edición de perfil: Permite al usuario modificar sus datos personales, enlaces y foto de perfil.

A continuación, se desglosan los requisitos funcionales y se responden los puntos solicitados.

a) Validación de entradas

En la vista de edición de perfil, se deben realizar las siguientes validaciones para asegurar que los datos ingresados sean correctos:

1. URL de la foto de perfil:
 - a. Validar que sea una URL válida con un formato correcto.
 - b. Verificar que la URL apunte a un archivo de imagen (por ejemplo, .jpg, .png).
2. Nombre y apellido:
 - a. Asegurar que no estén vacíos.
 - b. Permitir solo caracteres alfabéticos (y espacios si es necesario).
3. Datos personales:
 - a. Titular: Validar que no esté vacío y que tenga una longitud máxima de 75 caracteres.
 - b. Afiliación: Asegurar que no esté vacío.
 - c. Descripción: Permitir texto con un límite máximo de 500 caracteres.

Enlaces:

Validar que sean URLs válidas o permitir que queden vacíos si el usuario no desea asignarlos.

Correo personal:

Validar que el texto ingresado sea un correo electrónico válido en formato usuario@dominio.com.

b) Secuencias de operación exactas

1. **Visualización del perfil:**
 - a. El usuario accede a la vista del perfil y visualiza su información personal.
 - b. Opciones disponibles:
 - i. Editar Perfil: Al hacer clic, se redirige al formulario de edición.
 - ii. Cerrar sesión: Termina la sesión del usuario y lo redirige a la pantalla de inicio o autenticación.
2. **Edición del perfil:**
 - a. El usuario accede al formulario de edición.
 - b. Modifica los campos deseados (datos personales, enlaces, foto de perfil).

- c. Hace clic en "Guardar cambios".
- d. El sistema valida las entradas y guarda los datos en la base de datos (si son válidos).
- e. El usuario es redirigido nuevamente a la vista del perfil con los datos actualizados.

c) Respuestas a situaciones anormales

1. Sobreflujo:

- Escenario: El usuario intenta ingresar datos excesivamente largos o inválidos.
- Respuesta esperada:
 - Limitar la longitud de los campos (ejemplo: descripción con un máximo de 500 caracteres).

Mostrar mensajes de error como: "El texto es demasiado largo. Redúzcalo a 50 caracteres."

2. Facilidades de comunicación:

Mostrar mensajes claros para informar al usuario si ocurre un error o si la operación fue exitosa:

- "Tus datos se han actualizado correctamente."
- "No se pudo guardar la información. Revisa los campos e inténtalo de nuevo."

3. Manipulación y recuperación de errores:

- "La URL debe ser válida y apuntar a una imagen."
- Error en el servidor: Mostrar un mensaje genérico, como:
 - "Hubo un problema al guardar los datos. Inténtalo más tarde."

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias:

- **Entrada:**
 - En el formulario de edición, el usuario introduce o actualiza los siguientes datos:
 - URL de la foto de perfil.
 - Nombre y apellido.
 - Título profesional.
 - Afiliación.
 - Descripción personal.

- Enlaces (LinkedIn, GitHub, CV, Correo personal).
- El usuario hace clic en "Guardar cambios".
- **Salida:**
 - Si la validación es exitosa, el sistema almacena los datos en la base de datos y redirige nuevamente a la vista de perfil con la información actualizada.
 - Si hay errores, el sistema muestra mensajes de error específicos junto a los campos incorrectos.

2. Fórmulas de conversión de entrada/salida:

- **Foto de perfil:**
 - Entrada: URL proporcionada por el usuario.
 - Salida: La URL se utiliza para cargar y mostrar la imagen en la vista de perfil.
 - Validación: Verificar que la URL apunte a un archivo de imagen válido.
- **Enlaces:**
 - Entrada: URLs proporcionadas para LinkedIn, GitHub, CV y correo personal.
 - Salida: Los enlaces se muestran como hipervínculos en la vista del perfil, o como "No asignado" si no se ingresó información.
- **Descripción personal:**
 - Entrada: Texto introducido por el usuario.
 - Salida: Se muestra truncado o completo en la vista del perfil, dependiendo del diseño.
- **Correo personal:**
 - Entrada: Dirección de correo electrónico válida.
 - Salida: Se muestra como texto o hipervínculo en la vista del perfil.

Seccion Coaches:

El cliente menciona brevemente que deseaba un sitio donde se pudieran apreciar todos los expertos en el área de a modo de catálogo, después de esta breve definición concluimos que en dicha "Vista" existían 2 requisitos primarios:

1. Visualización de perfiles de usuario

- a. Cada usuario tiene un nombre, un título académico o profesional, y una afiliación.
- b. Se proporciona un botón "Ver Perfil" para acceder a más detalles.

2. Estructura de datos para usuarios

- a. Cada usuario tiene los atributos: nombre completo, título y afiliación.
- b. Opcionalmente, puede incluir una imagen de perfil.

Nota: Esta vista únicamente será de visualización (ya que sería ilógico modificar los datos de los otros perfiles)

a) Validación de entradas:

Esta sección formalmente no aplica, ya que, al ser una ventana de visualización, formalmente no se le ofrece al sistema ninguna entrada, a excepción de cuando se desea visualizar con mayor detalle el perfil (Existe un botón denominado ver perfil), con el objetivo de apreciar los detalles de cada perfil específico.

b) Secuencias de operación exactas:

1. El usuario accede a la lista de perfiles.
2. La información básica de cada usuario se muestra en tarjetas con:
 - a. Nombre completo.
 - b. Título académico/profesional.
 - c. Afiliación.
 - d. Imagen de perfil, si está disponible.
 - e. Botón "Ver Perfil".
3. Al hacer clic en "Ver Perfil", se redirige al detalle del perfil de ese usuario.

c) Respuestas a situaciones anormales:

1. **Sobreflujo:**
 - a. Si la cantidad de usuarios excede el espacio disponible para mostrarlos en una página, implementar paginación o scroll infinito.
2. **Facilidades de comunicación:**
 - a. Mostrar un mensaje de error claro si no se puede cargar la lista de perfiles (e.g., "No se pudo cargar la información. Intente de nuevo más tarde.").
 - b. Proveer un botón "Reintentar" para recargar los datos.
3. **Manipulación y recuperación de errores:**
 - a. Si algún dato no está disponible (e.g., título o afiliación), mostrar un valor predeterminado como "Información no disponible".
 - b. Si la imagen de perfil no está disponible, usar un ícono genérico por defecto.

d) Efectos de parámetros:

- La longitud del nombre o el título puede afectar el diseño visual (e.g., truncar texto si excede el espacio disponible).
- El tamaño o la resolución de la imagen de perfil puede afectar el rendimiento de la página si no se optimiza.

e) Relación de salidas con las entradas:

1. Entrada/Salidas de secuencias:

- a. Entrada: Datos del usuario desde la base de datos o una API (nombre, título, afiliación, imagen).
- b. Salida: Tarjetas renderizadas en la interfaz con la información correspondiente.

2. Fórmulas de conversión de entrada/salida:

- a. Si el nombre o título excede una longitud predeterminada, truncar con "..." para mantener el diseño limpio.
- b. Las imágenes deben redimensionarse (e.g., 250x250 píxeles) para un formato uniforme.

Importante:

- Implementar buenas prácticas de accesibilidad, como etiquetas alt para imágenes.

Ver perfil de un experto en el área:

En el requisito inmediato anterior mencionamos que, para cada experto en el área, es posible presionar el botón “Ver perfil”, al presionar dicho botón se te redirigirá a la página de perfil del dicho usuario, la cual es idéntica a tu propio perfil (Ver requisito “Ver mi perfil”), la única diferencia radica en que como obviamente no es tu perfil, no puedes modificar los datos de un perfil ajeno y menos cerrar su sesión. Excluyendo esas 2 funcionalidades, el requisito es idéntico a “Ver mi perfil”.

Mi Calendario:

La sección denominada “Mi calendario” debe constar de 3 componentes principales los cuales son:

- El botón de “Agregar Horario”
- La sección “Reuniones que ofrezco”
- La sección “Reuniones a las que estoy registrado”

Ofrecer reunión de entrevista:

Como se mencionó en un principio, los expertos en el área son capaces de ofrecer reuniones de asesoría/mentoría; a continuación se expone el flujo de oferta de reuniones.

Al presionar el boton "Agregar horario" (Presente en el calendario), se desplegara una ventana la cual preguntara por los datos: dia, hora de inicio y hora de fin. A continuacion proporcionaremos mayor contexto:

a) Validación de entradas

1. Validar que el campo "Día" sea una fecha válida y futura (o actual si aplica).
2. Validar que las "Horas de inicio" y "Horas de fin" tengan un formato correcto (HH:MM).
3. Validar que la "Hora de inicio" sea anterior a la "Hora de fin".
4. Prevenir el ingreso de caracteres no válidos (por ejemplo, letras en campos de tiempo).
5. Evitar conflictos de horarios ya registrados (no permitir solapamientos).

b) Secuencias de operación exactas

1. El usuario presiona el botón "+ **Agregar Horario**".
2. Se despliega el modal "Seleccionar Horario" con los siguientes campos:
 - a. Día.
 - b. Hora de inicio.
 - c. Hora de fin.
3. El usuario llena los campos correspondientes.
4. El usuario presiona el botón **Aceptar**:
 - a. Si los datos son válidos, se guarda el horario.
 - b. Si los datos son inválidos, se muestra un mensaje de error.
5. Si el usuario presiona **Cancelar**, se cierra el modal sin guardar cambios.

c) Respuestas a situaciones anormales

1. Sobreflujo

- Prevenir el ingreso de valores fuera de rango:
 - Fechas fuera del calendario válido (por ejemplo, 30/02/aaaa).
 - Horas fuera del rango 00:00 a 23:59.
- Limitar la cantidad máxima de horarios registrados (si aplica).

2. Facilidades de comunicación

- Mostrar mensajes claros en caso de errores, como:
 - "La hora de inicio debe ser anterior a la hora de fin".
 - "El horario seleccionado entra en conflicto con otro ya registrado".
 - "Por favor, introduzca una fecha válida".
- Utilizar indicadores visuales (por ejemplo, resaltar en rojo los campos con errores).

3. Manipulación y recuperación de errores

- Permitir al usuario corregir los campos sin cerrar el modal.
- En caso de error de validación, mantener los valores ingresados por el usuario para que no tenga que reescribirlos.

d) Efectos de parámetros

- El formato de fecha y hora debe ser configurable según la localización del usuario (por ejemplo, dd/mm/aaaa o mm/dd/aaaa).
- Si el sistema utiliza zonas horarias, ajustar los horarios ingresados al huso horario del usuario.

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias

- Entrada: El usuario ingresa la fecha, hora de inicio y hora de fin.
- Salida: El sistema registra el horario y lo muestra en la lista de "Reuniones que ofrezco".

2. Fórmulas de conversión de entrada/salida

- Convertir las horas ingresadas en un formato estándar (por ejemplo, UTC) para almacenamiento.
- Mostrar las horas almacenadas en el formato regional del usuario al desplegar la lista de reuniones.

Sección “Reuniones que ofrezco”:

Como lo indica la subsección el objetivo de este apartado es mostrar de manera estructurada y organizada las reuniones que el usuario ofrece, el cliente exige que se muestre como si fuera un calendario real, procederemos a especificar mejor esta sección.

a) Validación de entradas

1. Verificar que el usuario seleccione al menos un candidato antes de confirmar.
2. Verificar que los datos de los candidatos seleccionados (como correo electrónico) tengan un formato válido.
3. Validar que la hora de la reunión corresponda a la ya registrada (no debe permitir cambios fuera del horario programado).

b) Secuencias de operación exactas

1. Visualización de reuniones programadas:

- a. El usuario visualiza una lista de reuniones que ofrece, con fecha, horario y el número de candidatos asociados.
- b. Puede elegir entre:
 - i. Ver candidatos seleccionados (botón verde).
 - ii. Seleccionar candidatos para la reunión (botón azul).

2. Ver detalles de la reunión:

- a. Al hacer clic en "Ver candidatos seleccionados", se abre un modal con:
 - i. Fecha, hora de inicio y fin de la reunión.
 - ii. Enlace a la reunión.
 - iii. Detalles de los candidatos seleccionados (correo, motivo, tipo de reunión).

Nota: Esta opción solo aparece si ya se han seleccionado candidatos a entrevistar.

3. Seleccionar candidatos:

- a. Al hacer clic en "Seleccionar Candidato", se despliega un modal donde:
 - i. Se muestran los datos de la reunión programada (fecha y hora).
 - ii. El usuario puede seleccionar a él o los candidatos que desee entrevistar.
 - iii. Al presionar **Confirmar**, los candidatos seleccionados se asocian a la reunión.
- b. Si el usuario presiona **Cancelar**, se cierra el modal sin guardar cambios.

c) Respuestas a situaciones anormales

1. Sobreflujo

- Limitar el número máximo de candidatos que pueden ser seleccionados para una reunión (si aplica).
- Evitar duplicados en los candidatos seleccionados para la misma reunión.

2. Facilidades de comunicación

- Mostrar mensajes claros en caso de errores, como:
 - "Debe seleccionar al menos un candidato".
 - "El campo [Tipo/Motivo/Comentarios] no puede estar vacío".
 - "El formato del enlace a la reunión no es válido".
- Proveer retroalimentación visual sobre los campos con errores (por ejemplo, resaltarlos en rojo).

3. Manipulación y recuperación de errores

- Permitir al usuario corregir errores en el modal de selección sin necesidad de cerrarlo.
- En caso de fallo al guardar la información (por problemas de red, por ejemplo), mostrar un mensaje para reintentar.

d) Efectos de parámetros

1. El enlace de la reunión es un parámetro crítico. Si no es funcional o está mal ingresado, los asistentes no podrán acceder. (Lo genera automáticamente el sistema)
2. Los comentarios de los candidatos podrían influir en la decisión del usuario al momento de seleccionarlos.
3. Los cambios en la hora de una reunión ya registrada podrían afectar a los candidatos seleccionados (por ejemplo, si ya no pueden asistir en el nuevo horario).

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias

- **Entradas:**
 - Horario de la reunión.
 - Enlace de la reunión.
 - Selección de candidatos (Tipo, Motivo, Comentarios).
- **Salidas:**
 - Detalles de la reunión programada: Fecha, hora

Reuniones a las que estoy registrado:

Como su nombre lo indica el objetivo de esta sección es de proveer feedback al estudiante sobre cuales reuniones se encuentra inscrito, se deberá presentar como si se tratase de un calendario, a continuación, ofreceremos más detalles.

a) Validación de entradas

1. Validar que las reuniones mostradas en esta sección correspondan exclusivamente a aquellas en las que el usuario está postulado.
2. Validar que se actualice correctamente el estado de la postulación (por ejemplo, "En espera", "Aceptado", "Rechazado", etc.) según corresponda.
3. Asegurarse de que las reuniones pasadas no aparezcan en la lista, a menos que se requiera un historial de reuniones.

b) Secuencias de operación exactas

1. El sistema carga y muestra un listado de reuniones en las que el usuario está postulado.
 - a. Cada reunión incluye:
 - i. Fecha y hora.
 - ii. Nombre del anfitrión o responsable de la reunión (por ejemplo, "**Reunión con Kao Pech**").
 - iii. Estado de la postulación.
 - iv. Si el estado de la postulación se encuentra en espera o rechazado, no se mostrará el enlace de la reunión, pero, si es aceptado se le proveerá de la URL de la reunión.

c) Respuestas a situaciones anormales

1. Sobreflujo

- Si un usuario está postulado a muchas reuniones, mostrar un límite de reuniones por página y habilitar paginación o scroll infinito para manejar la carga de datos.

2. Facilidades de comunicación

- Mostrar estados claros y comprensibles para el usuario:
 - **"En espera"**: Postulación pendiente de revisión.
 - **"Aceptado"**: El anfitrión aceptó la postulación.
 - **"Rechazado"**: La postulación fue rechazada.
 - Otras opciones si son necesarias, como **"Finalizada"** o **"Cancelada"**.
- Proveer retroalimentación visual si ocurre un error al cargar la lista (por ejemplo, "No se pudieron cargar las reuniones. Por favor, intenta nuevamente").

3. Manipulación y recuperación de errores

- Si ocurre un fallo en la carga de reuniones por problemas de red o servidor, permitir al usuario reintentar la operación (por ejemplo, con un botón de **"Reintentar"**).
- Mostrar un mensaje como "No hay reuniones en las que estés postulado" si la lista está vacía.

d) Efectos de parámetros

1. El estado de la postulación afecta directamente la percepción del usuario sobre su participación en la reunión. Por ejemplo:
 - a. **"En espera"**: Indica que el anfitrión aún no ha tomado una decisión.
 - b. **"Aceptado"**: Confirma que el usuario asistirá a la reunión.
 - c. **"Rechazado"**: Notifica al usuario que no fue seleccionado.
2. La fecha y hora de la reunión determinan si debe mostrarse en esta lista (solo reuniones futuras, salvo que se requiera mostrar un historial).

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias

- **Entradas:**
 - Información almacenada sobre las reuniones en las que el usuario está postulado, incluyendo:
 - Fecha y hora.
 - Nombre del anfitrión.
 - Estado de la postulación.
- **Salidas:**
 - Lista de reuniones que cumplen con los criterios (futuras y en estado de postulación).
 - Estado asociado a cada reunión, mostrado de forma clara y comprensible.

2. Fórmulas de conversión de entrada/salida

- Convertir la información almacenada en la base de datos a un formato adecuado para la interfaz de usuario:
 - Formato de fecha y hora según la localización del usuario.
 - Estados de postulación en un lenguaje claro (por ejemplo, **"En espera"**).

Registrarse a una reunión

Como se ha mencionado a lo largo del documento, un estudiante al dirigirse al calendario de un experto puede postularse a una reunión generada por el experto, procederemos a especificar mejor dicho detalle.

a) Validación de entradas

1. Validar que el usuario complete todos los campos requeridos en el formulario de inscripción:
 - a. "Tipo de la Reunión".
 - b. "Motivo de la Reunión".

2. Permitir que el campo "Comentarios Adicionales" sea opcional.
3. Asegurarse de que el usuario no pueda registrarse en una reunión en la que ya está registrado.
4. Verificar que la fecha y hora de la reunión no hayan pasado antes de permitir el registro.

b) Secuencias de operación exactas

1. El usuario visualiza un calendario con reuniones disponibles. Cada reunión muestra:
 - a. Fecha y hora.
 - b. Número actual de participantes.
 - c. Un botón **Postularme** si el usuario no está registrado.
2. El usuario hace clic en **Postularme**:
 - a. Se abre un modal titulado "Inscribirme a la Reunión".
 - b. El usuario llena los campos requeridos: "Tipo de la Reunión", "Motivo de la Reunión" y opcionalmente "Comentarios Adicionales".
 - c. El usuario presiona **Aceptar** para confirmar su inscripción o **Cancelar** para salir sin guardar.
3. Si los datos son válidos, se muestra un mensaje de éxito: **"Te has inscrito exitosamente"**.
4. La interfaz se actualiza automáticamente:
 - a. El botón **Postularme** se reemplaza por un texto que indica **"Ya registrado"**.
 - b. El número de participantes de la reunión se incrementa en uno.

c) Respuestas a situaciones anormales

1. Sobreflujo

- Prevenir registros duplicados, si el usuario ya está registrado, deshabilitar el botón **Postularme** o mostrar un mensaje de error.
- Asegurarse de que los campos no excedan un límite de caracteres (por ejemplo, 250 caracteres para comentarios).

2. Facilidades de comunicación

- Mostrar mensajes claros en caso de errores, como:
 - "Ya estás registrado en esta reunión".
 - "Por favor, completa todos los campos obligatorios".
 - "Esta reunión ya no está disponible".
- Usar retroalimentación visual para indicar los campos con errores (por ejemplo, resaltarlos en rojo).

3. Manipulación y recuperación de errores

- Si ocurre un error al enviar los datos (por ejemplo, problemas de red), mostrar un mensaje como: "Ocurrió un error. Por favor, intenta nuevamente".
- Mantener los datos ingresados por el usuario en el formulario para evitar que los tenga que reingresar.

d) Efectos de parámetros

1. La fecha y hora de la reunión determinan si el registro está habilitado (no debe permitir registros en reuniones pasadas). (No debería mostrarlas)
2. Los valores ingresados por el usuario ("Tipo", "Motivo", "Comentarios") pueden influir en el proceso de aprobación o gestión posterior de la reunión.

e) Relación de salidas con las entradas

1. Entrada/salidas de secuencias

- **Entradas:**
 - Datos ingresados por el usuario en el formulario:
 - "Tipo de la Reunión".
 - "Motivo de la Reunión".
 - "Comentarios Adicionales" (opcional).
- **Salidas:**
 - Confirmación de que el usuario ha sido registrado en la reunión.
 - Actualización en la interfaz:
 - El botón **Postularme** cambia a **Ya registrado**.
 - El contador de participantes de la reunión se incrementa.

2. Fórmulas de conversión de entrada/salida

- No se identifican fórmulas matemáticas de conversión en este contexto. Sin embargo, se deben realizar conversiones internas, como:
 - Almacenar los datos ingresados en un formato adecuado para la base de datos.
 - Mostrar los datos en la interfaz en un formato comprensible para el usuario:
 - Por ejemplo, fechas y horas en el formato regional del usuario.

Desempeño de requerimientos (requerimientos no funcionales).

1. Rendimiento:

- a. El sistema debe procesar las solicitudes en menos de 2 segundos bajo una carga de 100 usuarios simultáneos.

- b. El tiempo de respuesta no debe exceder los 5 segundos en situaciones de carga máxima, Esto aplica especialmente a los calendarios y perfiles.

2. **Escalabilidad:**

- a. El sistema debe ser capaz de manejar un crecimiento en el volumen de datos del 200% en los próximos 3 años. (Si es que asignan más presupuesto)

3. **Disponibilidad:**

- a. El sistema debe estar disponible el 85% del tiempo durante el horario laboral.
- b. Las interrupciones programadas para mantenimiento no deben exceder las 2 horas al mes.
- c. Se debe sincronizar y gestionar reuniones de forma precisa para evitar conflictos de horarios.

4. **Seguridad:**

- a. Información sensible como nombres, afiliaciones y enlaces debe estar protegida frente a accesos no autorizados.
- b. Se debe incluir mecanismos para garantizar que solo usuarios autorizados puedan acceder, editar su perfil y participar en reuniones.
- c. Garantizar que ciertos datos, como el correo personal y CV, solo sean visibles si el usuario lo asigna.

5. **Usabilidad:**

- a. La interfaz debe ser intuitiva y fácil de usar para personas sin conocimientos técnicos.
- b. Los usuarios deben ser capaces de completar una tarea típica en menos de 5 pasos.
- c. La navegación entre perfiles, edición de datos y gestión de reuniones es sencilla y clara, lo que implica un diseño amigable para los usuarios.
- d. Botones como "Ver Perfil", "Editar Perfil", "Ver Calendario", "Seleccionar Candidato" y "Postúlate" son visibles y están etiquetados claramente.
- e. Los usuarios pueden ver y editar su información personal, como nombre, afiliación, descripción, y enlaces, de manera fácil y directa.

6. **Compatibilidad:**

- a. El sistema debe ser compatible con los navegadores modernos como Chrome, Firefox y Safari.
- b. Debe ser accesible desde dispositivos móviles y de escritorio.

7. **Mantenibilidad:**

- a. El código fuente debe ser modular y seguir estándares establecidos como SOLID.

- b. Debe incluirse documentación técnica completa para facilitar futuras modificaciones.
- c. El sistema debe permitir la adición de nuevas funcionalidades o corrección de errores sin afectar el funcionamiento actual.
- d. Uso de estándares de desarrollo que faciliten la comprensión y modificación por parte de los desarrolladores.

8. **Confiabilidad:**

- a. El sistema debe poder recuperarse automáticamente de fallos menores en menos de 20 minutos.

9. **Portabilidad:**

- a. El sistema debe poder migrarse a otro entorno de servidor en menos de 7 días.

10. **Accesibilidad:**

- a. Los usuarios con discapacidades deben poder navegar y utilizar el sistema sin barreras.
- b. El diseño debe considerar opciones para usuarios con discapacidades, como contraste adecuado, textos descriptivos y navegación mediante teclado.

11. **Internacionalización:**

- a. El sistema debe soportar múltiples idiomas, incluyendo inglés y español.
- b. El sistema debe ser capaz de adaptar los horarios de las reuniones a un horario local.

14. **Interactividad:**

- a. Los estados de las reuniones y postulaciones deben actualizarse inmediatamente para reflejar la información más reciente.