
Informe de Revisión de Sprint y Estado del Proyecto, Arquitectura y Diseño

Consortio de Gobiernos Autónomos Provinciales del Ecuador

Dirección de Tecnologías e Información

PROYECTO:

**“Implementación de un sistema provincial integral para la gestión de los
GAD’s”**

Fecha Inicio Sprint: 11 de marzo del 2024

Fecha Fin Sprint: 28 de marzo del 2024

Versión 1.0
Fecha de emisión: 28 de marzo de 2024
Quito – Ecuador

Contenido

1.	ANTECEDENTES.....	3
2.	DESARROLLO	5
2.1.	ANÁLISIS DEL SISTEMA ACTUAL	5
2.1.1.	MODULOS	5
2.2.	PROPUESTA.....	6
2.2.1.	BASE DE DATOS	6
2.2.2.	BACK END	6
2.2.3.	FRONT END	7
2.2.4.	APLICACIÓN EN LA NUBE	7
2.3.	MODULOS	8
2.4.	ARQUITECTURA Y DISEÑO	9
2.4.1.	FUNCIONALIDAD DEMO POR CAPAS	9
2.5.	METODOLOGIA DE TRABAJO	12
3.	CONCLUSIONES Y RECOMENDACIONES	13
4.	ANEXOS.	¡Error! Marcador no definido.
5.	FIRMAS.....	14

1. ANTECEDENTES

- El artículo 148 del Reglamento a la Ley Orgánica del Servicio Público, establece: “Art. 148.- De los contratos civiles de servicios.-La autoridad nominadora podrá suscribir contratos civiles de servicios profesionales o contratos técnicos especializados sin relación de dependencia, siempre y cuando la UATH justifique que la labor a ser desarrollada no puede ser ejecutada por personal de su propia entidad u organización, fuere insuficiente el mismo o se requiera especialización en trabajos específicos a ser desarrollados, que existan recursos económicos disponibles en una partida para tales efectos, que no implique aumento en la masa salarial aprobada, y que cumpla con los perfiles establecidos para los puestos institucionales y genéricos correspondientes. Estos contratos se suscribirán para puestos comprendidos en todos los grupos ocupacionales y se pagarán mediante honorarios mensualizados.- Las personas a contratarse bajo esta modalidad no deberán tener inhabilidades, prohibiciones e impedimentos establecidos para las y los servidores públicos. Tratándose de personas que hayan recibido indemnización o compensación económica por compra de renuncia, retiro voluntario, venta de renuncia u otras figuras similares, no constituirá impedimento para suscribir un contrato civil de servicios, conforme lo establece la LOSEP y este Reglamento General.- (...)”.
- El CONGOPE, entidad asociativa de los gobiernos autónomos provinciales de carácter nacional, de derecho público con personería jurídica, autonomía administrativa y financiera y patrimonio propio, autoriza la contratación civil de servicios profesionales de conformidad con el informe constante en el memorando No. CONGOPE-DA-2024-0207-M de 04 de marzo de 2024 de la Dirección Administrativa del CONGOPE para la contratación civil de servicios profesionales de conformidad con el contenido de los memorandos de la Dirección de Tecnologías e Información del CONGOPE Nro. CONGOPE-DTICS-2024-0022-M y Nro. CONGOPE-DTICS-2024-0046-M de 29 de enero de 2024 y 04 de marzo del mismo año, respectivamente. La Directora Ejecutiva autorizó proceder de acuerdo con el respectivo recorrido del sistema Quipux.
- Las actividades que desempeñará el profesional contratado son específicas relacionadas al proyecto "Implementación de un sistema provincial integral para la gestión de los GAD's", de conformidad al requerimiento presentado por el Director de Tecnologías e Información, en los términos de referencia, y que se detallan en el numeral "3. Responsabilidades y Tareas del Profesional", de dichos términos.
- El profesional deberá cumplir las siguientes responsabilidades y tareas:
 - Participar en el proceso de actualización tecnológica del sistema SFGProv.
 - Desarrollar y supervisar la implementación de nuevas funcionalidades y mejoras que se realicen al sistema.

- Colaborar estrechamente con el equipo existente, aportando experiencia técnica y contribuyendo a la integración de tecnologías actuales.
 - Elaborar manuales técnicos que permiten una transferencia de conocimiento transparente de lo implementado.
- La profesional generará los siguientes productos:
 - 1. Documentos de Requerimientos de Módulos:
Historias de Usuario: Desglose de los requisitos en historias de usuario con criterios de aceptación, priorizados por el Product Owner para la implementación del “Sistema Provincial Integral para la gestión de los Gobiernos Autónomos Descentralizados Provinciales”.
 - 2. Informes de Avances:
Informe de Revisión de Sprint: Actualización al final de cada sprint que incluye el progreso realizado, las historias de usuario completadas y cualquier impedimento.
 - 3. Código del Módulo / Aplicativo:
Entregas Incrementales: Proporcionar entregas incrementales al final de cada sprint, con código funcional y comentarios sobre posibles mejoras.
 - 4. Manuales Técnicos:
Documentación de Arquitectura y Diseño: Descripción de la arquitectura y diseño del sistema, actualizada al final de cada sprint según los cambios realizados.
 - 5. Manuales de Usuario:
Guía de Usuario Actualizada: Manual del usuario que refleja las funcionalidades implementadas en el sprint más reciente.
- **PLAZO**
La duración del presente contrato es de diez (10) meses calendario contados a partir de su suscripción.

Con fecha 11 de marzo de 2024, se firmó el contrato de servicios profesionales cuyo objetivo es contar con una profesional que desempeñe el puesto de Analista de Proyectos de Desarrollo de Software del “Proyecto de Implementación del Sistema Provincial Integral para la gestión de los Gobiernos Autónomos Descentralizados Provinciales (GADP)”

2. DESARROLLO

2.1. ANÁLISIS DEL SISTEMA ACTUAL

Actualmente el sistema tiene una estructura monolítica el mismo que maneja una base PostgreSQL y aplicación Visual Basic 6.

La base de datos que actualmente utiliza la aplicación ha tenido un procedimiento de actualización el cual permite tener una base de datos consistente y con los parches de seguridad que recomienda el fabricante.

La aplicación actualmente tiene un nivel de madurez estable, sin embargo, está desarrollada en Visual Basic 6.0 que es un lenguaje de programación para Windows que aún es utilizado. Sin embargo, Microsoft dejó de dar soporte oficial a Visual Basic 6.0 en 2008. Esto significa que no hay actualizaciones de seguridad, garantía de que el software funcione correctamente en versiones nuevas, no es posible seguir con el mantenimiento de la aplicación y se limita su uso en distribuciones que sean propias de Windows.

Por lo antes expuesto el sistema no es de fácil mantenimiento y su estructura se vuelve compleja a medida del crecimiento del sistema no es escalable.

Tomando en consideración el plazo del contrato y los productos a ser realizados se realiza el siguiente análisis con respecto a la estructura del sistema a desarrollarse.

2.1.1. MODULOS

El sistema cuenta con los siguientes módulos

- AUTENTICACIÓN Y PERFILES
- PARAMETRIZACION DEL SISTEMA
- CONTABILIDAD
- PRESUPUESTO
- INVENTARIOS
- NOMINA
- ACTIVOS FIJOS
- TESORERIA
- Módulos de acceso de acuerdo con perfiles o actividades extras
- GARANTÍAS
- CONTROL DE OBRAS
- CONTROL DE BIENES INFORMÁTICOS
- RECAUDACIÓN
- Módulos en la Web
- EXTENSIONES TELEFÓNICAS

- NÓMINA WEB
- CONSULTAS CONTABLES
- CONSULTAS PRESUPUESTARIAS

2.2. PROPUESTA

2.2.1. BASE DE DATOS

La base de datos se mantendrá en postgreSql y se utilizará para el desarrollo la última versión estable, a la fecha de este informe la versión 16.2.

La estructura de la base de datos mantendrá su diseño actual, sin embargo, si se requieren realizar modificaciones las mismas se realizarán en estrecha coordinación con el equipo técnico de la Dirección de Tecnologías e información.

2.2.2. BACK END

El sistema tendrá una arquitectura de capas, evitando que se conecte directamente los datos a la aplicación, motivo por el cual se propone desarrollar servicios API REST.

Las API REST se comunican mediante solicitudes HTTP para realizar funciones de bases de datos estándar como crear, leer, actualizar y eliminar registros (también conocidas como CRUD) dentro de un recurso. Por ejemplo, una API REST usaría una solicitud GET para recuperar un registro, una solicitud POST para crearlo, una solicitud PUT para actualizar un registro y una solicitud DELETE para eliminarlo. Todos los métodos HTTP se pueden utilizar en llamadas a API. Una API REST es similar a un sitio web que se ejecuta en un navegador web con funcionalidad HTTP incorporada.

La tecnología plantada para el desarrollo del backEnd es C#, esta tecnología fue considerada debido a que ASP.NET facilita la creación de servicios que lleguen a una amplia gama de clientes, incluidos exploradores y dispositivos móviles. Los puntos de conexión serializan automáticamente las clases en JSON con el formato correcto de forma predeterminada. La protección de las conexiones API se realiza con JSON estándar del sector. En resumen, se recomienda el uso de C# como la opción más adecuada para desarrollar API REST por ofrecer seguridad, soporte multiplataforma, mayor rendimiento, diversidad de recursos de apoyo disponibles y facilidad de desarrollo. A continuación, se definen algunas de estas ventajas

Seguridad: C# y la plataforma .NET proporcionan un conjunto robusto de características de seguridad integradas que pueden ayudar a proteger las API REST contra amenazas como inyecciones de código SQL, ataques de scripting entre sitios (XSS) y otros ataques comunes a la seguridad web.

Amplia adopción: Hay una gran cantidad de recursos, documentación y comunidad de desarrolladores disponibles para ayudar en el desarrollo de API REST en C#.

Rendimiento: C# es un lenguaje que se ejecuta en el entorno de tiempo de ejecución de .NET, que está optimizado para un alto rendimiento. Esto lo hace adecuado para aplicaciones que requieren una alta capacidad de respuesta y eficiencia en el manejo de solicitudes HTTP.

Facilidad de desarrollo: C# es un lenguaje moderno y orientado a objetos que ofrece una sintaxis clara y legible. Además, la plataforma .NET proporciona una amplia variedad de bibliotecas y marcos de trabajo que facilitan el desarrollo de API REST. ASP.NET Core es especialmente adecuado para la creación de servicios web escalables y de alto rendimiento.

Soporte multiplataforma: Con ASP.NET Core es posible desarrollar aplicaciones web y API REST en C# que se pueden ejecutar en sistemas operativos diferentes de Windows, como Linux y macOS, lo que aumenta la flexibilidad y la portabilidad de las aplicaciones.

2.2.3. FRONT END

Se realizará la migración de los formularios disponibles actualmente en Visual Basic 6 a Windows Form Versión .NET 8.0, IDE: Visual Studio Comunity 2022.

2.2.4. APLICACIÓN EN LA NUBE

Se recomienda el alojamiento del aplicativo en un entorno virtual Cloud Computing por los siguientes aspectos:

Acceso remoto: La nube te permite acceder a tus archivos y aplicaciones desde cualquier lugar con conexión a internet, lo que es especialmente útil para personas que trabajan desde casa, viajan con frecuencia o necesitan acceder a su información mientras están en movimiento.

Escalabilidad: Con la nube, puedes escalar fácilmente tus recursos informáticos según tus necesidades. Si tu empresa experimenta un aumento repentino en el tráfico o la demanda de almacenamiento, puedes aumentar la capacidad de manera rápida y sencilla sin tener que invertir en hardware adicional.

Costos reducidos: En lugar de invertir en infraestructura física costosa, como servidores y equipos de almacenamiento, puedes pagar solo por los recursos que utilizas en la nube. Esto puede reducir significativamente los costos operativos y eliminar la necesidad de mantenimiento y actualizaciones de hardware.

Seguridad: Los proveedores de servicios en la nube suelen ofrecer medidas de seguridad avanzadas para proteger tus datos, como cifrado de datos, firewalls y monitoreo continuo. Esto puede proporcionar un nivel de seguridad más alto que el que muchas empresas pueden lograr por sí mismas.

Colaboración: La nube facilita la colaboración entre equipos distribuidos geográficamente. Varios usuarios pueden trabajar en documentos compartidos en tiempo real, lo que mejora la eficiencia y la productividad.

Respaldo y recuperación de datos: Los servicios en la nube suelen ofrecer funciones de respaldo automáticas y redundancia de datos, lo que garantiza que tus archivos estén protegidos contra pérdidas y sean recuperables en caso de desastre.

En resumen, utilizar la nube puede ofrecer flexibilidad, eficiencia, seguridad y ahorro de costos para la organización.

La aplicación podría ser implementada en servidores proporcionados por la institución y en equipos de escritorio para despliegues donde no se tenga alta de concurrencia de usuarios

2.3. MÓDULOS

Para la base del sistema se desarrollarán los siguientes módulos

- AUTENTICACIÓN Y PERFILES
- PARAMETRIZACION DEL SISTEMA

De acuerdo con la construcción actual del sistema, se consideran para el desarrollo de la solución los siguientes módulos:

- CONTABILIDAD
- PRESUPUESTO
- INVENTARIOS
- NOMINA
- ACTIVOS FIJOS
- TESORERIA

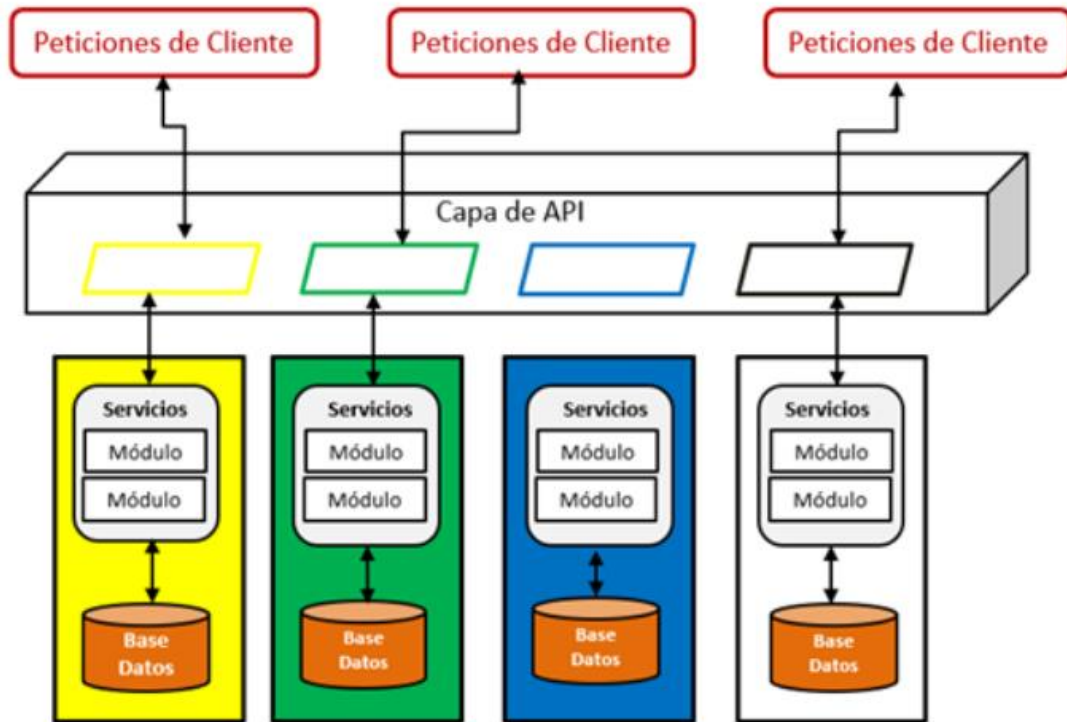
El desarrollo de los módulos se realizará de acuerdo con la prioridad establecida a la interna de la institución.

2.4. ARQUITECTURA Y DISEÑO

2.4.1. FUNCIONALIDAD DEMO POR CAPAS

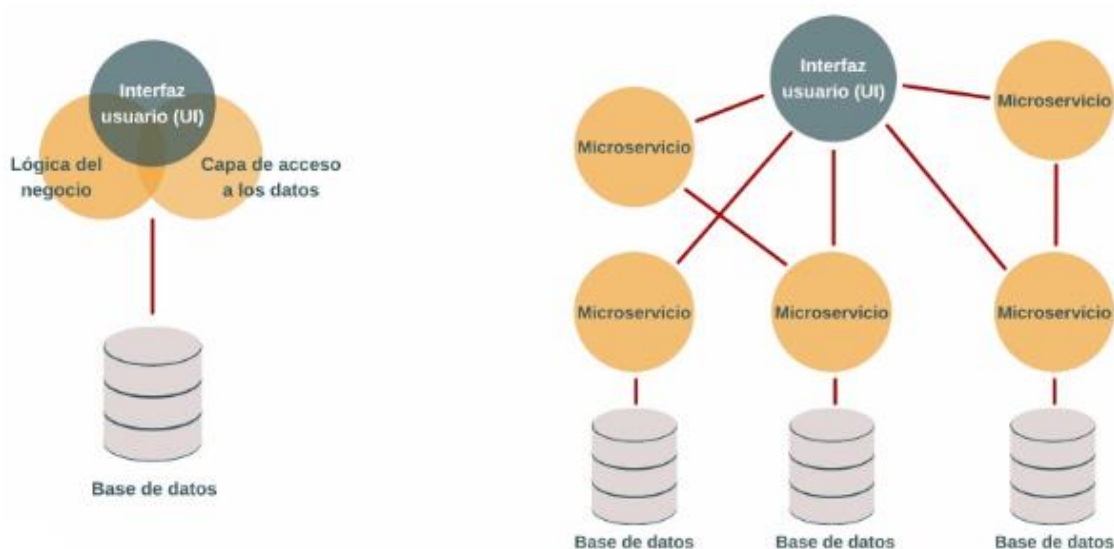
Se realiza la comparación de arquitecturas monolítica y de microservicios.

Microservicios



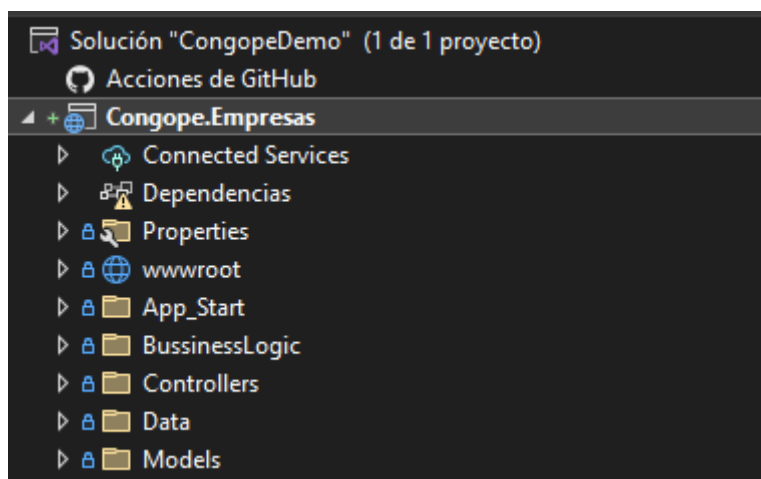
Fuente: <https://www.linkedin.com/pulse/patrones-en-la-arquitectura-de-software-elmo-renato-castro-ramirez/?originalSubdomain=es>

Arquitectura monolítica vs arquitectura de microservicios



Fuente: <https://click-it.es/arquitectura-monolitica-vs-arquitectura-de-microservicios-cual-debo-elegir/>

Para el back – end se creó la aplicación por capas de Lógica de Negocio, **Controladores y Modelos**



Modelos

```

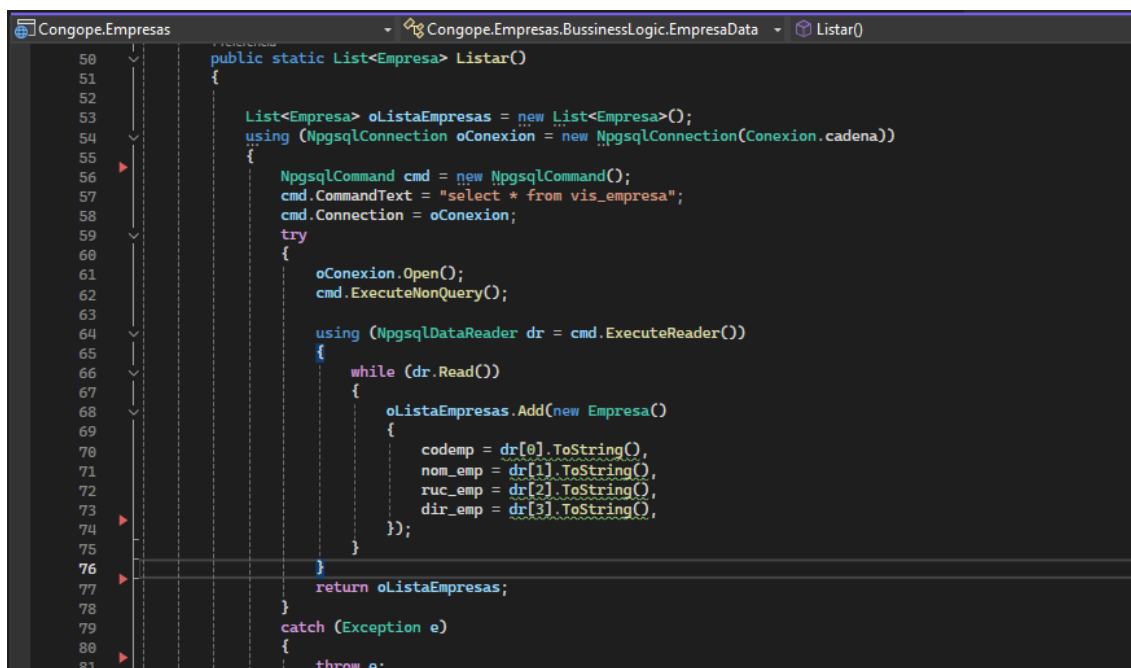
Congope.Empresas  Congope.Empresas.Models.Empresa
1 namespace Congope.Empresas.Models
2 {
3     14 referencias
4     public class Empresa
5     {
6         4 referencias
7         public string codemp { get ; set ; }
8         4 referencias
9         public string nom_emp { get ; set; }
10        2 referencias
11        public string ruc_emp { get; set; }
12        1 referencia
13        public string nomcomercial { get; set; }
14        2 referencias
15        public string dir_emp { get; set; }
16        1 referencia
17        public string tel1_emp { get; set; }
18        1 referencia
19        public string tel2_emp { get; set; }
20        1 referencia
21        public string fax1_emp { get; set; }
22    }
23 }
    
```

Controladores

```

Congope.Empresas  Congope.Empresas.Controllers.EmpresaControlle  Get(string id)
1 using Congope.Empresas.BussinessLogic;
2 using Congope.Empresas.Models;
3 using Microsoft.AspNetCore.Mvc;
4
5     /// For more information on enabling Web API for empty projects, visit https://go.microsoft.com/fwlink/?LinkId=301871
6
7 namespace Congope.Empresas.Controllers
8 {
9     [Route("api/[controller]")]
10    [ApiController]
11    0 referencias
12    public class EmpresaController : ControllerBase
13    {
14        // GET: api/<ValuesController>
15        [HttpGet]
16        0 referencias
17        public List<Empresa> Get()
18        {
19            return EmpresaData.Listar();
20        }
21
22        // GET api/<ValuesController>/5
23        [HttpGet("{id}")]
24        0 referencias
25        public Empresa Get(string id)
26        {
27            return EmpresaData.Obtener(id);
28        }
29
30        /// GET api/<ValuesController>/6
31    }
32 }
    
```

Lógica de Negocio



```
50 public static List<Empresa> Listar()
51 {
52
53     List<Empresa> oListaEmpresas = new List<Empresa>();
54     using (NpgsqlConnection oConexion = new NpgsqlConnection(Conexion.cadena))
55     {
56         NpgsqlCommand cmd = new NpgsqlCommand();
57         cmd.CommandText = "select * from vis_empresa";
58         cmd.Connection = oConexion;
59         try
60         {
61             oConexion.Open();
62             cmd.ExecuteNonQuery();
63
64             using (NpgsqlDataReader dr = cmd.ExecuteReader())
65             {
66                 while (dr.Read())
67                 {
68                     oListaEmpresas.Add(new Empresa()
69                     {
70                         codemp = dr[0].ToString(),
71                         nom_emp = dr[1].ToString(),
72                         ruc_emp = dr[2].ToString(),
73                         dir_emp = dr[3].ToString(),
74                     });
75                 }
76             }
77             return oListaEmpresas;
78         }
79         catch (Exception e)
80         {
81             throw e;
82         }
83     }
84 }
```

2.5. METODOLOGIA DE TRABAJO

- Para el avance del proyecto se especifica utilizar la metodología SCRUM, en la que es imperioso nombrar un funcionario como (Product Owner, QA) para la gestión general del proyecto.
- El Product Owner, será quien priorice el desarrollo de los módulos de acuerdo con su conocimiento y quien finalmente valide el producto entregado por el equipo contratado.
- Se han considerado sprints de 3 semanas que permitan realizar un seguimiento adecuado de actividades que deben cumplirse para llegar a las metas.
- Los desarrolladores de la institución (3 recursos humanos) no dedicarán el 100% de su tiempo al proyecto debido al soporte a sistemas existentes y tareas propias de sus cargos, sin embargo, es importante su participación en el proyecto debido al conocimiento del funcionamiento del sistema.
- Cada módulo es distinto al uno del otro, y las funcionalidades tienen distinta complejidad, motivo por el cual, se tendrá que realizar ajustes al cronograma y eso podría conllevar a que no sea posible cumplir con los tiempos inicialmente considerados. Este desglose proporciona un panorama detallado de las actividades específicas que se llevarán a cabo en cada sprint. Esto facilitará el seguimiento y la gestión efectiva del progreso del proyecto.

Por cada sprint se levantará documentación que respalde el trabajo ejecutado en el mismo es decir historias de usuario, diagramas de diseño de arquitectura, manuales de usuario y técnicos y demás documentación necesaria para que el proyecto lleve un control documental adecuado.

Las capacitaciones al personal institucional de tecnologías de información, se realizará de acuerdo con la aplicación de las herramientas durante el desarrollo de la solución.

Cabe mencionar que semanalmente se está realizando la entrega de la planificación correspondiente.

3. CONCLUSIONES Y RECOMENDACIONES

- Se recomienda que las personas de las diferentes áreas designen un responsable para que desempeñe su papel como Product Owner dando la priorización a los módulos que serán objeto del desarrollo.
- La tecnología utilizada puede ser desplegada en una infraestructura en la nube, en una infraestructura in house, o una máquina de escritorio, como recomendación es implementarla en una infraestructura de nube tomando en consideración las características tecnológicas que esta ofrece, se debe realizar un adecuado dimensionamiento de capacidades y de costos previa la implementación.
- La arquitectura propuesta permite que el equipo de trabajo pueda dividir sus actividades de desarrollo de acuerdo con la expertiz que maneje evitando de esta forma duplicar esfuerzos en actividades similares.
- La propuesta planteada presenta una arquitectura por capas en lenguaje C#, como ide de desarrollo se utilizará Visual Studio Community 2022, Framework net8.0 y para el front se realizará la migración de los formularios disponibles actualmente en visual basic Windows Form.
- Durante la última reunión del sprint se dio por aprobada la estructura planteada por el equipo contratado y así mismo se plantea el porcentaje de participación del equipo interno para el desarrollo del sistema SFGProv, a lo que mencionaron, que posiblemente seria del 50% del horario laboral.

4. FIRMAS

	CARGO	FIRMA	FECHA
ELABORADO POR	Norma Arcos Profesional Contratado		28-03-2024
	Javier Chillogallo Profesional Contratado		28-03-2024
REVISADO POR	Lenin Illapa Analista Informático		28-03-2024
APROBADO POR	Fidel Acero Director de Tecnologías e Información		28-03-2024