

## Actividad 1

### Actividad 1 - Creación del documento inicial del proyecto

Mauricio Bello Cogollo

Jhonny Alejandro Sánchez Quintana

Ingeniería de Software.

29082022\_C12\_202234: Proyecto de software

Magíster: Johan Manuel Gordillo Mesa

30 septiembre de 2022

Actividad 1

AUTORES

MAURICIO BELLO COGOLLO

JHONNY ALEJANDRO SÁNCHEZ QUINTANA

Presentado para optar el título de: Ingeniero de Software.

PROFESOR

MAGÍSTER: JOHAN MANUEL GORDILLO MESA

CORPORACIÓN UNIVERSITARIA IBEROAMERICANA

FACULTA DE INGENIERÍA

CARRERA DE INGENIERÍA DE SOFTWARE

Bogotá D.C

21 septiembre de 2022.

## Contenido

Planteamiento.....	<b>Error! Bookmark not defined.</b>
Introducción .....	5
Justificación .....	6
Objetivos .....	7
Objetivo General .....	7
Objetivos Específicos.....	7
1.    Planteamiento Del Proyecto .....	8
1.1.    Problema .....	8
2.    Alcance.....	9
2.1.    Desarrollo De Un Sistema De Información Para La Gestión Y Alquiler De Salones (Dynamic Services).....	9
2.1.1.    Capa de Configuracion y seguridad: .....	9
2.1.2.    Capa de Negocio .....	9
2.1.3.    Capa de Datos .....	10
2.1.4.    Capa del lado del cliente .....	10
3.    Estructura Arquitectonca.....	10
Análisis de riesgo.....	12
Roles .....	12
Planificación de sprints .....	13
Glosario.....	20
Definición: EndPoints.....	20
Definicion: Error.....	20

Definicion: Defecto.....	20
Definicion: Fallo .....	20
Bibliografía .....	21

## Introducción

Cuando identificamos la solución a un problema tenemos el 50% del todo, faltándonos así el otro 50%; y es lo que plantaremos en el presente trabajo, a continuación, podremos identificar un problema de negocio al cual aplicaremos una solución concreta, solución detallada así: especificaremos un proyecto de software para la renta de salones para eventos especiales, aplicando estándares de calidad y seguridad que garanticen el crecimiento económico óptimo para los clientes que lo adquieran.

## Justificación

El rápido avance de las nuevas tecnologías las cuales fomentan la innovación y el interés por mantenerse competitivo, ha llevado progresivamente al desarrollo de nuevas formas de recopilar datos, de una manera más precisa y eficiente, obligando a las pequeñas, medianas y grandes empresas, a contratar o crear soluciones informáticas personalizadas que se ajusten a su modelo de negocio, permitiendo así mejorar sus procesos internos, satisfacer las necesidades de sus clientes y obtener un gran incremento económico y seguridad en sus productos. El proceso de calidad y seguridad bien aplicado garantiza bajos costos, “la prevención siempre es mejor que la reparación”, además de que conlleva a muchos beneficios; tales como, buena fama, productos confiables, escalables, eficientes etc.

## Objetivos

### Objetivo General

Desarrollar una aplicación web con estándares de calidad y seguridad con el fin de gestionar la renta de salones para eventos especiales tales como: bodas, reuniones de negocio, entrevistas, y demás eventos.

### Objetivos Específicos

Crear un módulo llamado Personas, para controlar quien renta y a quien se le renta cada salón.

Crear un módulo de Seguridad para controlar los permisos de quien renta y a quien se le renta cada salón.

Crear un módulo de Balanceo para controlar los recursos y centralizar la seguridad de la aplicación.

Crear un módulo transversal para centralizar el flujo de los datos de la aplicación.

Crear un módulo para realizar la reserva y renta de los salones.

## 1. Planteamiento Del Proyecto

### 1.1. Problema

En la ciudad de bogota existen pequeños micro empresarios atascados tecnológicamente del sector de **renta de salones** (salones para fiestas, reuniones, bodas, cumpleaños, lugares para eventos especiales, iglesias y demas), normalmente ellos ofresen sus servicios a traves del periodico local, bolantes, pregoneo en la calle o recomendaciones de sus anteriores clientes, ademas que la negociacion siempre es de palabra y con apuntes en un cuaderno, dando pie para cometer multiples errores como: agendar dos veces el mismo salón, realizar una mala cotizacion etc, no siendo del todo malo pero si muy ineficiente y poco productivo; una solucion es: crear una aplicación para realizar todas estas actividades de forma interactiva, facil y segura.



## 2. Alcance

Descripcion del proyecto de software sobre el cual se desarrollará el entregable.

### 2.1. Desarrollo De Un Sistema De Información Para La Gestión Y Alquiler De Salones (Dynamic Services)

Dynamic Services, es un proyecto de software con el objetivo de construir un producto que permita la administración e integración de diferentes tipos de módulos y para diferentes sectores económicos, no obstante, para el entregable académico se pretende desarrollar un módulo de renta o alquiler de **Salones para eventos**. Este software contiene validación de seguridad por capas, y cada uno de sus módulos esta sementado por microservicios REST independientes, cada uno tiene control de autenticación y autorización con Token web JSON. La aplicación contara con los siguientes Microservicios, librerias y aplicaciones web:

#### 2.1.1. Capa de Configuracion y seguridad:

##### 2.1.1.1. DynamicServices.Gateway

Este API se encargara de balacear y centralizar todos los metodos de los diferentes servicio de la aplicación a traves de Ocelote.

##### 2.1.1.2. DynamicServices.Security

Este API se encargara de controlar la generacion de token de acceso para los deferentes microservicios.

#### 2.1.2. Capa de Negocio:

##### 2.1.2.1. DynamicServices.ComunES

Este API se encargara de controlar la generacion de metodos globales (Global EndPoints)

##### 2.1.2.2. DynamicServices.PersonasES

Este API se encargara de controlar la generacion de EndPoints para la gestion de los difentes tipos de persona, (clientes, usuarios, adminstrador, persona natural, empresas)

#### 2.1.2.3. DynamicServices.SalonesES

Este API se encargara de controlar la generacion de EndPoints para la gestion o renta de salose

#### 2.1.3. Capa de Datos

##### 2.1.3.1. DynamicServices.GlobalModel

Esta librería se encarga de agrupar las clases canonicas y de base de datos para uso trasversal del y serializacion de los datos.

#### 2.1.4. Capa del lado del cliente

##### 2.1.4.1. DynamicServices.Web

Esta APP WEB se encargara de la capa del lado del cliete, Front end.

### 3. Estructura Arquitectonca.

La arquitectura sera tipo cliente servidos en capas a traves de microservicios REST FULL siguiendo el siguinete patron global.

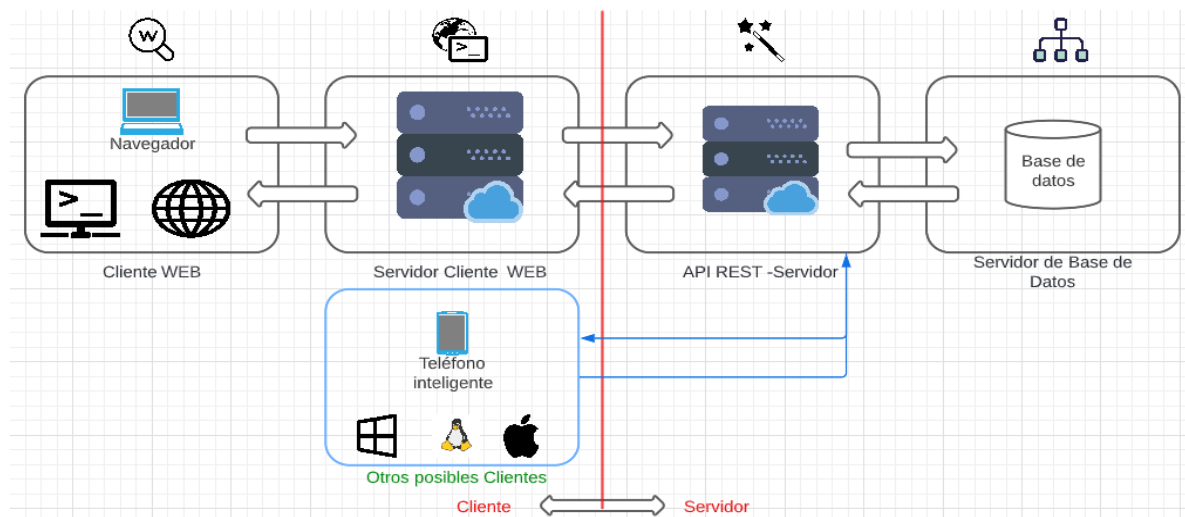


Figura 1.

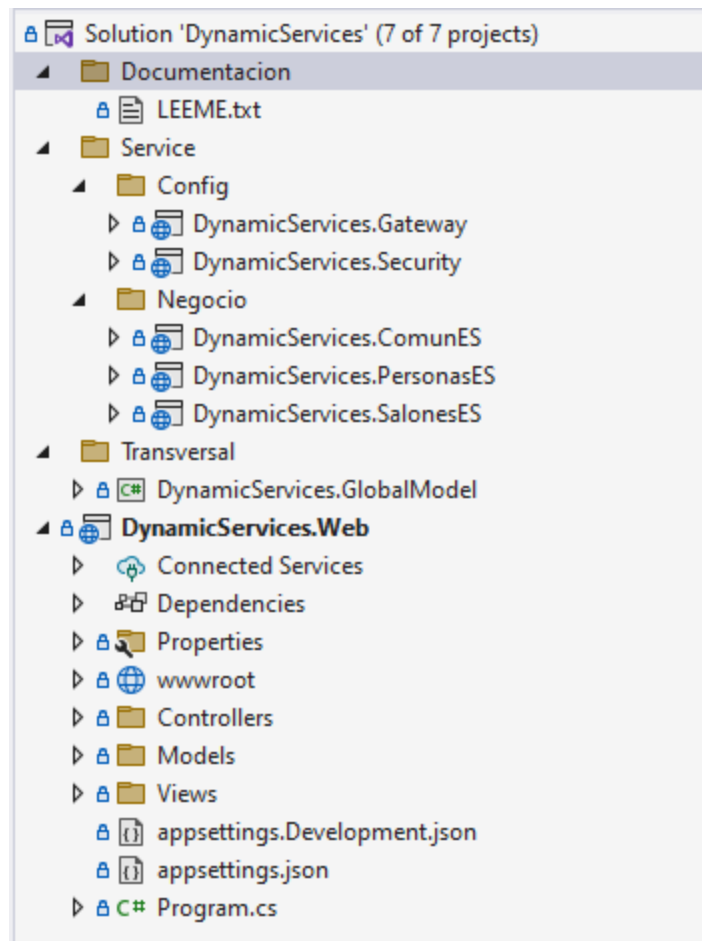


Figura 2

## Análisis de riesgo

### Evaluación

ID LINK	DEFINICIÓN	CONTROLES	PROBABILIDAD DE OCURRENCIA	IMPACTO	RIESGO
<a href="#">1D</a>	Definición del alcance	Especificación de requerimientos	BAJO	ALTO	MEDIO
<a href="#">2D</a>	Diseño MER	Diseño del flujo de datos	BAJO	ALTO	ALTO
3D	Codificación	Desarrollo prototipo	ALTO	MEDIO	MEDIO
4D	Amenazas	No identificada	BAJO	BAJO	BAJO
5D	Activos	Personal de desarrollo y herramientas	BAJO	BAJO	BAJO
6D	Vulnerabilidad	No identificada	BAJO	BAJO	BAJO

### Roles

CARGO	ENCARGADO	RESPONSABILIDAD
Arquitecto	Mauricio Bello	Diseñar los artefactos a desarrollar, especificar la arquitectura utilizada
Analista	Jhonny Alejandro Sánchez	Levantamiento de requerimientos, investigación de evaluación de riesgos,
Desarrollador Front-End	Jhonny Alejandro Sánchez	Desarrollo web interfaz grafica
Desarrollador Back-End	Mauricio Bello	Desarrollo código fuente microservicios
Analista de pruebas	Jhonny Alejandro Sánchez	Realizar pruebas funcionales
Scrum Master	Jhonny Sánchez	Organizar cronograma de actividades

## Planificación de sprints

En esta fase, se realizó una estimación del tiempo de duración de cada una de las historias de usuarios, pertenecientes a los procesos que conforma el sistema de control.

Tabla 1: Estimación de Historia de Usuarios Administrativos

*Fuente: Autor*

Proceso	No	Historia de usuario	TIEMPO ESTIMADO		
			SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS
Personal Administrativo	1	Registro de salóns	0.4	2	8
	2	Registro de arrendatarios	0.4	2	8
	3	Registro de arrendador	0.4	2	8
	4	Registro de contratos de arrendamiento	0.4	2	8
	5	Renovación de contratos de arrendamiento	0.4	2	8

\

	<b>6</b>	Enrolamiento y registro de usuarios	1	5	8
	<b>TIEMPO ESTIMADO</b>		3	15	48

Tabla 2: Estimación de Historia de Usuarios Arrendatarios

*Fuente: Autor*

<b>Proceso</b>	<b>No</b>	<b>Historia de usuario</b>	<b>TIEMPO ESTIMADO</b>		
			<b>SEMANAS ESTIMADAS</b>	<b>DIAS ESTIMADOS</b>	<b>HORAS ESTIMADAS</b>
Arrendatarios	<b>1</b>	Registro de salón	0.4	2	8
	<b>2</b>	Revisión estado salón	0.4	2	8
	<b>TIEMPO ESTIMADO</b>		0.8	4	16

Tabla 3: Estimación de Historia de Usuarios Arrendador

*Fuente: Autor*

Proceso	No	Historia de usuario	TIEMPO ESTIMADO		
			SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS
Arrendador	1	Consultar disponibilidad de salón	0.4	2	8
	2	Agendar cita de visita a salón	0.4	2	8
	3	Arrendar salón	0.4	2	8
		<b>TIEMPO ESTIMADO</b>	1.2	6	24

## 1. Plan de Entrega

Tabla 4: Plan de entrega (a)

*Fuente: Autor*

Proceso	Nº	Historia de usuario	TIEMPO ESTIMADO			Iteración asignada				Entrega asignada			
			SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS	1	2	3	4	1	2	3	4
Personal Administrativo	1	Registro de salóns	0.4	2	8	X				X			
	2	Registro de arrendatarios	0.4	2	8	X					X		
	3	Registro de arrendador	0.4	2	8	X				X			
	4	Registro de contratos de arrendamiento	0.4	2	8	X						X	



	<b>5</b>	Renovación de contratos de arrendamiento	0.4	2	8	X					X		
	<b>6</b>	Enrolamiento y registro de usuarios	1	5	8	X						X	

Tabla 5: Plan de entrega (b)

*Fuente: Autor*

Proceso	Nº	Historia de usuario	TIEMPO ESTIMADO			Iteración asignada				Entrega asignada			
			SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS	1	2	3	4	1	2	3	4
Arrendatarios	<b>1</b>	Registro de salón	0.4	2	8				X		X		
	<b>2</b>	Revisión estado salón	0.4	2	8				X			X	

\

--	--

Tabla 6: Plan de entrega (c)

*Fuente: Autor*

Proceso	Nº	Historia de usuario	TIEMPO ESTIMADO			Iteración asignada				Entrega asignada			
			SEMANAS ESTIMADAS	DÍAS ESTIMADOS	HORAS ESTIMADAS	1	2	3	4	1	2	3	4
Arrendador	1	Consultar disponibilidad de salón	0.4	2	8			X			X		
	2	Agendar cita de visita a salón	0.4	2	8			X					X
	3	Arrendar salón	0.4	2	8			X				X	

	<b>TIEMPO ESTIMADO</b>	6.2	25	88	
--	----------------------------	-----	----	----	--

## Glosario

### Definición: EndPoints

EndPoints ( Punto final de API), Un punto final de API es un punto en el que una API, el código que permite que dos programas de software se comuniquen entre sí, se conecta con el programa de software. Las API funcionan enviando solicitudes de información desde una aplicación web o un servidor web y recibiendo una respuesta.[5]

### Definicion: Error

Es la acción humana que genera un incorrecto resultado, una idea equivocada de algo. Define la equivocación del desarrollador o del analista.

Ejemplos de errores pueden ser:

Error de lógica de programación: este produce un fallo posterior al desarrollo de la aplicación. Error de sintaxis: este produce un fallo inmediato en el código fuente.

Un requerimiento mal especificado.[6]

### Definicion: Defecto

Se encuentra en algún componente del sistema. Imperfección de un componente causado por un error. El analista es quien debe reportar el defecto al ejecutar los casos de prueba.

Ejemplos de defecto pueden ser:

Un módulo de registro de usuarios tiene mal el método de encriptar contraseña. [6]

### Definicion: Fallo

Es la manifestación visible de un defecto. Es decir que si un defecto es encontrado durante la ejecución de una aplicación entonces va a producir un fallo. Generación de un mensaje de alerta que no fue contemplado previamente por el desarrollador, una excepción no controlada. [6]

## Bibliografía

[1] Bourque, P. & Fairley, R. E. (2014). Chapter 13. Computing Foundations. 17. Secure Software Development and Maintenance. En Swebok. Guide to the Software Engineering Body of Knowledge - Version 3.0. (pp. 13-24 / 13-27). IEEE Computer Society.

[2] Piattini Velthuis, M. G. García Rubio, F. O. y García Rodríguez de Guzmán, I. (2018). Calidad de sistemas de información (4a. ed.). RA-MA Editorial. Recuperado de: <https://elibro.net/en/lc/biblioibero/titulos/106509>.

[3] Sanchez, S., Sicilia, M. & Rodriguez, D. (2012). Ingeniería del software. Un enfoque desde la guía SWEBOK. Alfaomega. Recuperado de: <https://www-alphaeditorialcloud-com.ibero.basesdedatosezproxy.com/reader/ingenieria-del-software-un-enfoque-desde-la-guia-swebok?location=4>.

[4] Moreno J. J., Bolaños, L. P., & Navia, M. A. (2010). Exploración De Modelos Y Estándares De Calidad Para El Producto Software. UIS Ingenierías, 9(1), 39–53. Recuperado de: <https://search-ebscohost-com.ibero.basesdedatosezproxy.com/login.aspx?direct=true&db=a9h&AN=55346741&lang=es&site=ehost-live>.

[5] <https://www.techtarget.com/searchapparchitecture/definition/API-endpoint#:~:text=An%20API%20endpoint%20is%20a,server%20and%20receiving%20a%20response.>

[6] <https://gbitcorp.com/blog/posts/error-defecto-y-fallo/> [Pagina 1].

\

[7]Esterkin, V., & Pons, C. (2017). Evaluación de calidad en el desarrollo de software dirigido por modelos. INGENIARE - Revista Chilena de Ingeniería, 25(3), 449–463. .

Recuperado de: <https://search-ebscohost->

[com.iberobasesdedatosezproxy.com/login.aspx?direct=true&db=a9h&AN=125137510&lang=es](https://search-ebscohost-com.iberobasesdedatosezproxy.com/login.aspx?direct=true&db=a9h&AN=125137510&lang=es)  
[&site=ehost-live.](#)

[https://gbitcorp.com/blog/posts/error-defecto-y-fallo/.](https://gbitcorp.com/blog/posts/error-defecto-y-fallo/)