

Actividad 2

Construir en el documento la planeación 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.

Creación del documento inicial del proyecto

CONSTRUIR EN EL DOCUMENTO LA PLANEACIÓN DEL PROYECTO

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

PROYECTO DE SOFTWARE

Bogotá D.C

30 Septiembre de 2022.

JUEGO DE ROLES

Contenido	
Planteamiento.....	5
Según las necesidades del proyecto y por el número de gestores de este, cada estudiante asumirá uno o más roles activos. (ejemplo, levantamiento de información, análisis, diseño, acercamientos con el cliente).	5
Introducción	6
Justificación	7
Objetivos	8
Objetivo General.....	8
Objetivos Específicos.....	8
1. Planteamiento Del Proyecto.....	9
1.1. Problema	9
2. Alcance	10
2.1. Desarrollo De Un Sistema De Información Para La Gestión Y Alquiler De Salones (Dynamic Services).....	10
2.1.1. Capa de Configuración y seguridad:	10
2.1.2. Capa de Negocio	10
2.1.3. Capa de Datos	11
2.1.4. Capa del lado del cliente	11
3. Estructura Arquitectónica.....	11
Planificación de Sprints	12

Creación del documento inicial del proyecto

Análisis de riesgo	18
Matriz de riesgos.....	18
Roles	19
Bibliografía	23

JUEGO DE ROLES

Planteamiento

Según las necesidades del proyecto y por el número de gestores de este, cada estudiante asumirá uno o más roles activos. (ejemplo, levantamiento de información, análisis, diseño, acercamientos con el cliente).

Creación del documento inicial del proyecto

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 de eventos especiales, aplicando estándares de calidad y seguridad que garanticen el un crecimiento económico óptimo para los clientes que lo adquieran.

JUEGO DE ROLES

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.

Creación del documento inicial del proyecto

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.

JUEGO DE ROLES

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 ofrecen sus servicios a través del periodico local, bolantes, pregoneo en la calle o recomendaciones de sus anteriores clientes, además que la negociación siempre es de palabra y con apuntes en un cuaderno, dando pie para cometer múltiples errores como: agendar dos veces el mismo salón, realizar una mala cotización etc, no siendo del todo malo pero si muy ineficiente y poco productivo; una solución es: crear una aplicación para realizar todas estas actividades de forma interactiva, fácil y segura.

2. Alcance

Descripción 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 está sustentado por microservicios REST independientes, cada uno tiene control de autenticación y autorización con Token web JSON. La aplicación contará con los siguientes Microservicios, librerías y aplicaciones web:

2.1.1. Capa de Configuración y seguridad:

2.1.1.1. DynamicServices.Gateway

Este API se encargará de balancear y centralizar todos los métodos de los diferentes servicios de la aplicación a través de Ocelote.

2.1.1.2. DynamicServices.Security

Este API se encargará de controlar la generación de token de acceso para los diferentes microservicios.

2.1.2. Capa de Negocio:

2.1.2.1. DynamicServices.ComunES

Este API se encargará de controlar la generación de métodos globales (Global EndPoints)

2.1.2.2. DynamicServices.PersonasES

Este API se encargará de controlar la generación de EndPoints para la gestión de los diferentes tipos de persona, (clientes, usuarios, administrador, persona natural, empresas)

JUEGO DE ROLES

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.

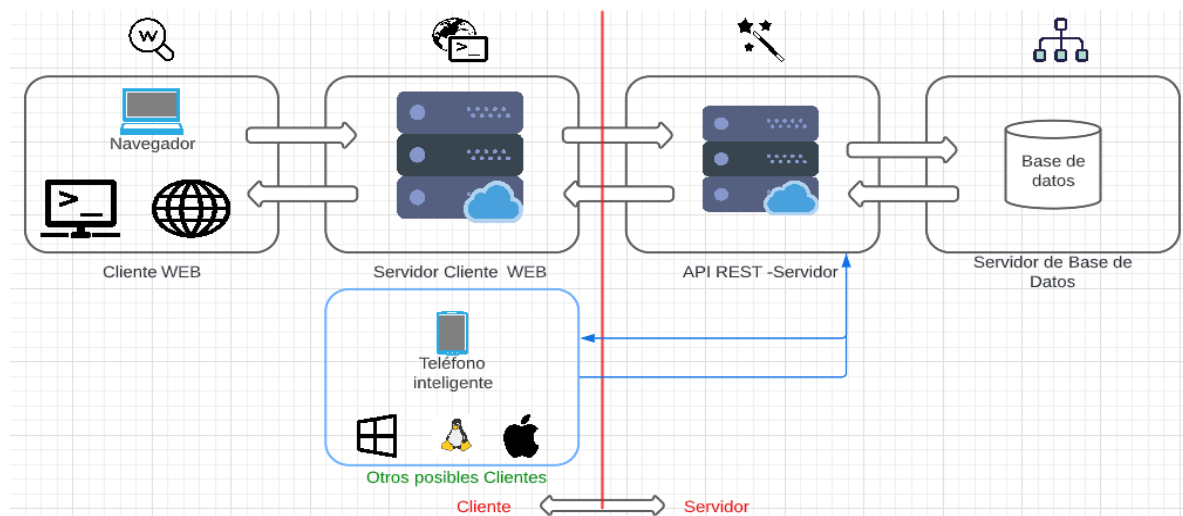


Figura 1.

Creación del documento inicial del proyecto

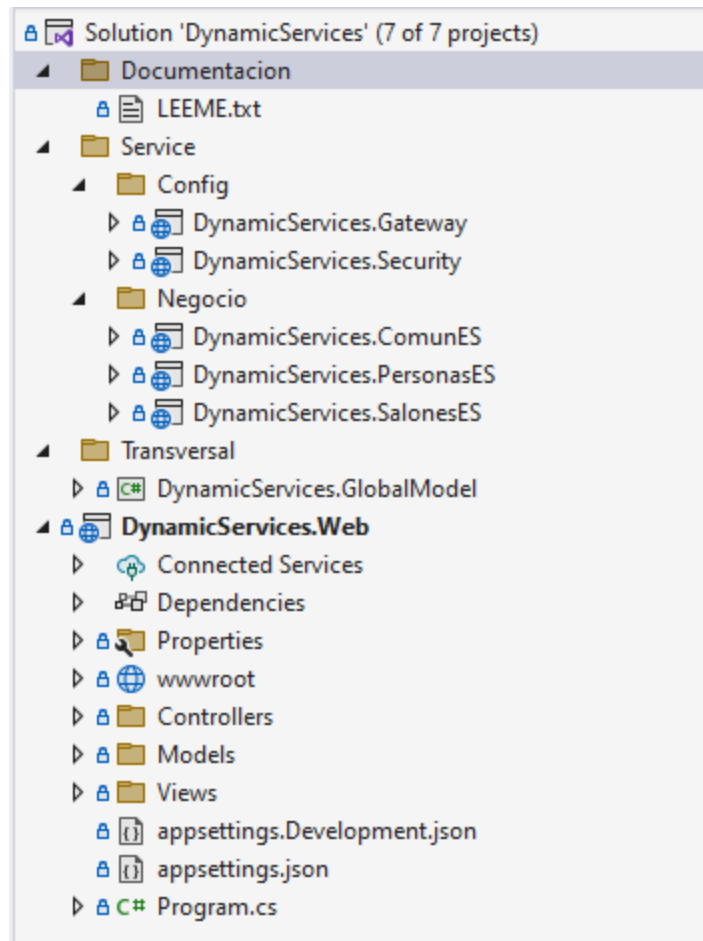


Figura 2.

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

JUEGO DE ROLES

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
	6	Enrolamiento y registro de usuarios	1	5	8
	TIEMPO ESTIMADO		3	15	48

Tabla 2: Estimación de Historia de Usuarios Arrendatarios

Fuente: Autor

Creación del documento inicial del proyecto

Proceso	No	Historia de usuario	TIEMPO ESTIMADO		
			SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS
Arrendatarios	1	Registro de salón	0.4	2	8
	2	Revisión estado salón	0.4	2	8
	TIEMPO ESTIMADO		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

JUEGO DE ROLES

	2	Agendar cita de visita a salón	0.4	2	8
	3	Arrendar salón	0.4	2	8
	TIEMPO ESTIMADO		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		

Creación del documento inicial del proyecto

3	Registro de arrendador	0.4	2	8	X					X			
4	Registro de contratos de arrendamiento	0.4	2	8	X							X	
5	Renovación de contratos de arrendamiento	0.4	2	8	X					X			
6	Enrolamiento y registro de usuarios	1	5	8	X							X	

Tabla 5: Plan de entrega (b)

Fuente: Autor

Proceso	Número	Historia de usuario	TIEMPO ESTIMADO			Iteración asignada				Entrega asignada			
			SEMANAS ESTIMADAS	DIAS ESTIMADOS	HORAS ESTIMADAS	1	2	3	4	1	2	3	4

JUEGO DE ROLES

Arrendatarios	1	Registro de salón	0.4	2	8				X			X	
	2	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	DIAS ESTIMADOS	HORAS ESTIMADAS	1	2	3	4	1	2	3	4
Arrendado	1	Consultar disponibilidad	0.4	2	8			X			X		

Creación del documento inicial del proyecto

		dad de salón												
	2	Agendar cita de visita a salón	0.4	2	8			X						X
	3	Arrendar salón	0.4	2	8			X					X	
	TIEMPO ESTIMADO		6.2	25	88									

Análisis de riesgo

Matriz de riesgos

ID LINK	DEFINICIÓN	CONTROLES	PROBABILIDAD DE OCURRENCIA	IMPACTO	RIESGO
<u>1D</u>	Definición del alcance	Especificación de requerimientos	BAJO	ALTO	MEDIO
<u>2D</u>	Diseño MER	Diseño del flojo de datos	BAJO	ALTO	ALTO
3D	Codificación	Desarrollo prototipo	ALTO	MEDIO	MEDIO
4D	Amenazas	No identificada	BAJO	BAJO	BAJO

JUEGO DE ROLES

5D	Activos	Personal de desarrollo y herramientas	BAJO	BAJO	BAJO
6D	Vulnerabilidad	No identificada	BAJO	BAJO	BAJO

Tabla 1.

Roles

RESPONSABILIDAD	A1	A2	A3
Desarrollador	Mauricio Bello Cogollo	Jhonny Alejandro Sánchez Quintana	Sin asignación
Front end	X	X	
Back end	X	X	
Documentación	X	X	
Levantamiento de Requerimiento		X	
Mockup		X	
Diseñar los artefactos a desarrollar, especificar la arquitectura utilizada	X		
Levantamiento de requerimientos, investigación de evaluación de riesgos,		X	
Desarrollo web interfaz grafica	X		
Analista de pruebas		X	
Coordinador Proyecto	X		
Realizar pruebas funcionales		X	
Desarrollo código fuente microservicios	X	X	
Arquitecto	X		

Tabla 2.

Planeación temporal (Cronograma y diagrama de gantt).

Anexo diagrama Gantt

Creación del documento inicial del proyecto

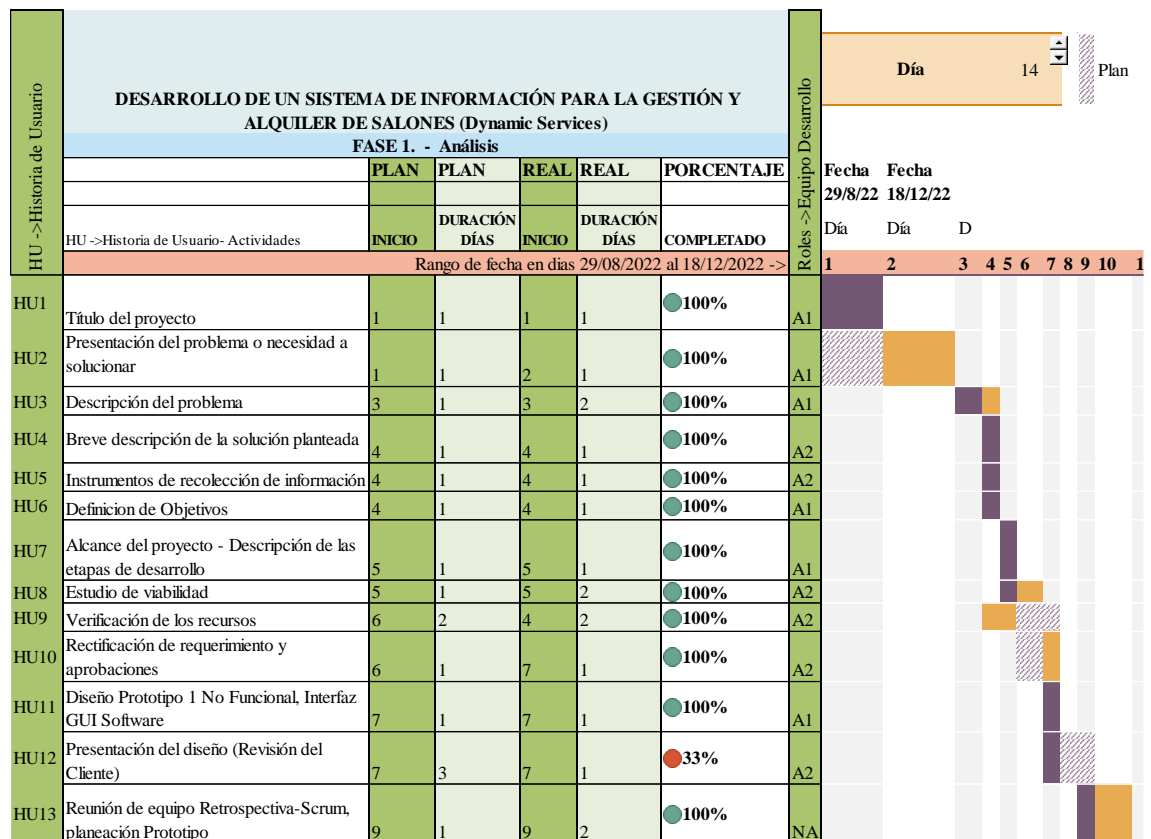


Figura 3.

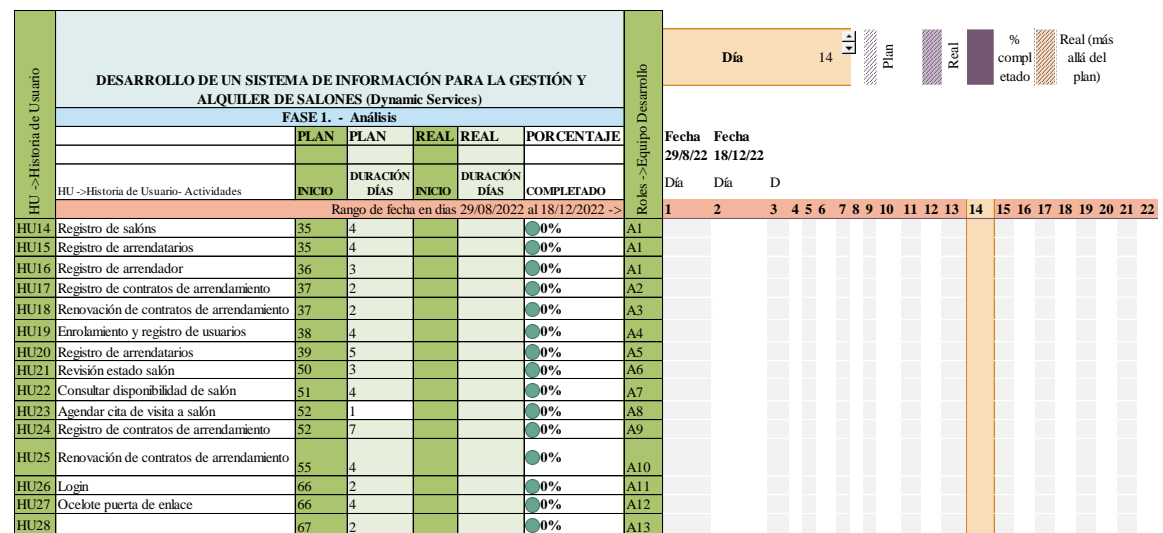


Figura 4.

Desglose de riesgos del proyecto (diagrama RBS)



Figura 5.

Conclusión

Si bien es cierto que identificar un problema permite actuar con precaución, determinar una solución adecuada es todo un reto. Desde el punto de vistas donde pretendemos sacar provecho, la identificación de una solución de software debe delimitar y precisar procesos coherentes para un público objetivo, no obstante un software que se construye con el fin de solucionar un problema específico, debe tener o contar con las características que le permitan avanzar, crecer y adaptarse a nuevas necesidades del usuario; algo constante es que el usuario siempre necesitar más... sin embargo un proyecto de software permite delimitar el alcance del mismo, este es el principal pilar para garantizar costes precisos y asequibles. Un equipo debe compararse a sí mismo, buscar la mejora continua y entregar más valor para el negocio, iteración tras iteración. Una buena planificación siempre trae buenos resultados y permite el cumplimiento de metas.

Bibliografía

Jones, C. (2008). Estimación de costos y administración de proyectos de software (2a. ed.).

Recuperado de la base de datos E-libro. Para consultarlo, revise la carpeta "Herramientas de apoyo" (Manual bibliotecas virtuales)

Gómez, R. J. (2016). Dirección y gestión de proyectos de tecnologías de la información en la empresa.

Recuperado de la base de datos E-libro. Para consultarlo, revise la carpeta "Herramientas de apoyo" (Manual bibliotecas virtuales)

[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>.

Creación del documento inicial del proyecto

[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.ibero.basesdedatosezproxy.com/login.aspx?direct=true&db=a9h&AN=125137510&lang=es&site=ehost-live>.

<https://gbitcorp.com/blog/posts/error-defecto-y-fallo/>.