

**DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE QUE PERMITA
GESTIONAR EL PORTAFOLIO DE SERVICIOS DEL GAME DEV LAB.**

EDGAR ANDRES GAMBOA SANABRIA

ISMAEL ANDRES PARRA PRATO

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

UNAB

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS

GRUPO DE INVESTIGACION EN TECNOLOGIAS DE INFORMACION

BUCARAMANGA

04/11/2020

**DESARROLLO DE UNA HERRAMIENTA DE SOFTWARE QUE PERMITA
GESTIONAR EL PORTAFOLIO DE SERVICIOS DEL GAME DEV LAB.**

EDGAR ANDRES GAMBOA SANABRIA

ISMAEL ANDRES PARRA PRATO

**TRABAJO DE GRADO PRESENTADO PARA OPTAR EL TÍTULO DE:
INGENIERO DE SISTEMAS**

PAULO CESAR RAMIREZ PRADA

INGENIERO DE SISTEMAS

MAGISTER EN GESTIÓN APLICACIÓN Y DESARROLLO DE SOFTWARE

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

UNAB

FACULTAD DE INGENIERÍA

INGENIERÍA DE SISTEMAS

BUCARAMANGA

04/11/2020

1 TABLA DE CONTENIDO

Contenido

1	TABLA DE CONTENIDO	4
2	LISTA DE TABLAS	6
3	LISTA DE FIGURAS	7
4	LISTA DE ANEXOS	8
5	RESUMEN	9
5.1	RESUMEN	9
5.2	PALABRAS CLAVE	9
5.3	LINEA DE INVESTIGACION	9
6	CAPÍTULOS	10
6.1	PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	10
6.1.1	PREGUNTA DE FONDO	11
6.2	OBJETIVOS	12
6.2.1	OBJETIVO GENERAL	12
6.2.2	OBJETIVOS ESPECÍFICOS	12
6.3	RESULTADOS ESPERADOS AL COMENZAR EL PROYETO	13
6.4	ANTECEDENTES	14
6.5	GLOSARIO	16
6.6	MARCO TEÓRICO	17
6.6.1	(CRS)COMPUTER RESERVATION SYSTEM	17
6.6.2	(XML) Extensive Markup Language	18
6.6.3	SOFTWARE DE GESTIÓN	19
6.6.4	MODELO C/S-B/S	20
6.6.4.1	C/S Cliente servidor	20
6.6.4.2	B/S Estructura de 3 capas	20
6.6.5	LABORATORIOS TRADICIONALES Y REMOTOS	21
6.7	ESTADO DEL ARTE	23
6.7.1	INTRODUCCIÓN	23
6.7.2	BADGER LAB	23
6.7.3	SAHARA	23
6.7.4	C/S-B/S Y LABORATORIOS ABIERTOS	26
6.7.5	DESARROLLO DISPONIBLE A LA VENTA	27
6.7.6	CHEDULING SYSTEM	28
6.8	SELECCIÓN DE HERRAMINETAS	30

6.8.1	APLICACIÓN WEB	30
6.8.2	APLICACIÓN MÓVIL	34
6.9	RESULTADOS OBTENIDOS	39
6.10	CONCLUSIONES	49
6.11	TRABAJOS A FUTURO	50
7	REFERENCIAS BIBLIOGRÁFICAS	53

2 LISTA DE TABLAS

Tabla 1. Resultados esperados.....	13
Tabla 2. Descripción Actor Usuario	39
Tabla 3. Descripción actor Tecnico de Laboratorio.....	39

3 LISTA DE FIGURAS

Fig.N.1 Sistema Sahara	26
Fig.N.2 Arquitectura NVLab	29
Fig.N.3 Lenguajes más usados	31
Fig.N.4 Frameworks más usados	32
Fig.N.5 Tamaño de paquetes angular	33
Fig.N.6 Tipos de aplicaciones móviles	34
Fig.N.7 Sistemas Operativos más usados.....	36
Fig.N.8 Lenguajes y Frameworks con mayor crecimiento	37
Figura 9. Arquitectura del sistema	40
Figura 10. Dummie echo a mano	41
Figura 11. Ventana de reserva	42
Figura 12. Diagrama de casos de uso.....	43
Figura 13. Diagrama de Clases.....	44
Figura 14. Carpetas del Proyecto	45
Figura 15. Servicio angularx social login	46
Figura 16. Inicio Sistema de Reservas	47
Figura 17. Pregunta de aceptación gráfica	48
Figura 18. Pregunta de aceptación lógica de los kits.....	48

4 LISTA DE ANEXOS

Anexo 1 (Documento de Análisis y Diseño).....	40
Anexo 2 (Dummies).....	40
Anexo 3 (Propuesta de Arquitectura Logica).....	42
Anexo 1 (Documento de Análisis y Diseño).....	44
Anexo 4 (Manual Técnico de usuario WEB).....	47
Anexo 5 (Resultados Pruebas de Aceptación Web).....	48
Anexo 6 (Plan de Trabajo).....	49

5 RESUMEN

5.1 RESUMEN

En el presente documento se encontrará información referente al proyecto: desarrollo de una herramienta de software que permita gestionar el portafolio de servicios del Game Dev Lab. Luego de ser creado el Laboratorio Game Dev Lab surgió la necesidad de crear un sistema de reservas que facilitara el almacenamiento, préstamo y reserva de los elementos del mismo y que funcionara en las plataformas web y móvil para mayor comodidad de los usuarios. El desarrollo del sistema fue llevado a cabo en 4 fases.

- Fase de Análisis.
- Fase de Diseño.
- Fase de Desarrollo.
- Fase de Pruebas de Aceptación.

Cada una de ellas permitió que el proyecto fuese llevado a cabo de una forma relativamente ordenada. Esto nos permitió identificar más aspectos de mejora los cuales serán llevados a cabo según el plan de trabajo a futuro.

5.2 PALABRAS CLAVE

API REST

APLICACIÓN WEB

COMPUTACIÓN EN LA NUBE

DESARROLLO DE SOFTWARE

SOFTWARE DE GESTIÓN

CRS (COMPUTER RESERVATION SYSTEM)

5.3 LINEA DE INVESTIGACION

SISTEMAS DE INFORMACIÓN E INGENIERÍAS DEL SOFTWARE

6 CAPÍTULOS

6.1 PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

A finales del 2017 la Universidad Autónoma de Bucaramanga (UNAB) creó un espacio en sus instalaciones llamado GAME DEV LAB, dicha zona cuenta con una vasta cantidad de elementos como: computadores de alto rendimiento, gafas VR, juegos de mesa, entre otros. Estos componentes están disponibles al público universitario, con el fin de darles un entorno adecuado para su desarrollo personal y académico.

El propósito del proyecto es desarrollar un software de gestión para la Game Dev Lab, el cual le facilite a los estudiantes y a la universidad el inventario, préstamo y reserva de los elementos allí presentes. Mediante funcionalidades, como: Registros virtuales, estadísticas de utilización, reportes de incidentes, caja de sugerencias entre otras.

Adicionalmente la facultad ingeniería de sistemas ha manifestado que el Game Dev Lab estará abierto de igual manera al público en general, esto para lograr una mejor integración con comunidades que sigan tendencias como la E-sport, computación gráfica, desarrollo de videojuegos entre otros, de acuerdo a lo establecido en el proyecto de investigación realizado por: paulo César Ramírez Prada, p. c. r. p., Nitae Andrés Uribe Ordoñez, n. a. u. o., & juan Sebastián cárdenas arenas, j. s. c. a. (2017).

Hoy en día el Game Dev Lab está siendo administrado de forma análoga, mediante registros impresos y correos que posteriormente son digitalizados en medios como Excel. La finalidad de este proceso es poder crear estadísticas, reportes de incidencias, historiales entre otros. Los cuales permiten saber el uso que se le está dando a la Game Dev Lab. Todo esto representa un trabajo arduo y paulatino que podría ser realizado de forma más eficaz si se implementara un CRS (Computer Reservation System).

Se podría decir que cualquier entidad que desee mantener un control sobre los elementos que son prestados. Está obligada a tener un CRS (Computer Reservation System) para optimizar su labor, como por ejemplo: La universidad San Buenaventura de Bogotá tiene un sistema llamado (audiovisuales) que administra la reserva de salas informáticas en el complejo educativo, Velandia, A. T., & Carlos, V. O. J. (2006). Por otra parte la biblioteca Dr. Antonio Sacoto ubicada en cantón biblián desarrolló una aplicación web que les permitió tener un control en el préstamo y consultas que se efectuaban dentro del recinto, Matamoros Motoche, M. A., & Sánchez Idrovo, M. F. (2011). Esto les dio la posibilidad de tener un registro preciso de sus actividades.

Así como este par de ejemplos hay muchos más en los cuales es evidente que la implementación de un CRS en entidades que lo requieran es una acción sensata ya que automatiza los procesos de forma eficaz, lo cual reduce costos y tiempo obviamente.

6.1.1 PREGUNTA DE FONDO

¿Cómo la Universidad Autónoma de Bucaramanga podrá implementar un CRS de tal forma que optimice los procesos de almacenamiento, préstamo y reserva en el Game Dev Lab?

6.2 OBJETIVOS

6.2.1 OBJETIVO GENERAL

- Desarrollo de una Herramienta Software Cliente-Servidor para Gestionar el Portafolio de Servicios ofertados a diferentes públicos objetivos por el Game Dev Lab, que Facilite El Almacenamiento, Préstamo Y Reserva De Elementos Del Laboratorio.

6.2.2 OBJETIVOS ESPECÍFICOS

- Analizar las necesidades del Game Dev Lab según el tipo de servicios que serán ofertados, los públicos objetivos de dichos servicios y las políticas institucionales de la UNAB.
- Diseñar las funcionalidades del software y las interfaces de usuario, de acuerdo a los requisitos funcionales y la arquitectura de software definida.
- Elaborar el software de acuerdo al diseño definido en una arquitectura orientada a la Web.
- Evaluar el correcto funcionamiento y el nivel de satisfacción de la herramienta software por medio de instrumentos de aceptación con los públicos objetivos.

6.3 RESULTADOS ESPERADOS AL COMENZAR EL PROYETO

Cada uno de estos resultados está basados en los objetivos específicos del proyecto los cuales fueron realizados satisfactoriamente. Posteriormente estos documentos y resultados se simplificaron y ayudaron a la elaboración de los anexos que se muestran en este documento como resultados obtenidos en el desarrollo del proyecto. *Tabla 1. Resultados esperados.*

OBJETIVOS	RESULTADOS ESPERADOS
Analizar las necesidades del Game Dev Lab según el tipo de servicios que serán ofertados, los públicos objetivos de dichos servicios y las políticas institucionales de la UNAB.	Documento de referentes de software para gestión de laboratorios
	Documento con el catálogo de servicios del Game Dev Lab
	Documento de requisitos funcionales de software
Diseñar las funcionalidades del software y las interfaces de usuario, de acuerdo a los requisitos funcionales y la arquitectura de software definida.	Documento de los procesos del Game Dev Lab
	Documento de diseño de interfaces de usuario
	Documento de diseño de software
Elaborar el software de acuerdo al diseño definido en una arquitectura orientada a la Web	Código fuente
	Diario de Desarrollo
	Manual Técnico y de Usuario
Evaluar el correcto funcionamiento y el nivel de satisfacción de la herramienta software por medio de instrumentos de aceptación con los públicos objetivos	Protocolo de pruebas
	Documento de recomendaciones y trabajo futuro

6.4 ANTECEDENTES

Se podría afirmar que el antecedente más notable de esta idea de desarrollo es el proyecto de investigación del Game Dev Lab paulo César Ramírez Prada, p. c. r. p., Nitae Andrés Uribe Ordoñez, n. a. u. o., & Juan Sebastián Cárdenas Arenas, j. s. c. a. (2017). En el cual se busca consolidar el Game Dev Lab mediante un software y un modelo de gestión sostenible, que permitan ofrecer un portafolio de servicios que integren los sectores empresarial, académico y comunidad Gamer. Este proyecto de investigación dio ciertas pautas para llegar a cumplir con los objetivos esperados. Como por ejemplo la arquitectura del proyecto, su alcance y reglas de negocio. A su vez el objetivo del proyecto será proveer la herramienta de software para la gestión de laboratorio, Se espera que el proyecto cumpla con estas expectativas y que a su vez contribuya con el crecimiento del Game Dev Lab.

Como referentes metodológicos se tiene a SIBU y el CPA. SIBU es el sistema de bibliotecas UNAB, este cuenta con un software robusto que le permite a la universidad tener control sobre el inventario, reservas, renovaciones y demás servicios que ofrecen las bibliotecas de nuestra institución, este sistema puede ser operado por estudiantes vía web o por los mismos funcionarios de las bibliotecas, cuando el estudiante decide hacer el proceso de préstamo directamente en la universidad. Por otra parte el CPA es el centro de producción audiovisual, una de las tantas responsabilidades de este departamento es administrar la aplicación web llamada Sistema de Reserva de Multimediales Audiovisuales, que es bastante conocida debido al constante uso por parte de los estudiantes cada vez que requieren de algún equipo audiovisual como por ejemplo, computadores cámaras tabletas entre otros.

Este par de departamentos no solo sirvieron como ejemplo en la metodología de sus sistemas de reserva sino que también han contribuido sus reglamentos como punto de referencia a la hora de crear uno propio para el Game Dev Lab, estos reglamentos se encuentran en el acuerdo número 103 expedido en marzo 6 del 2019, UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA. (2019, Marzo 6).

El Game Dev Lab es un laboratorio que servirá para labores académicas, de investigación y extensión al servicio del Programa de Ingeniería de Sistemas y la Facultad de Ingeniería, la cual cuenta con otros laboratorios como son: Laboratorio de Automatización, Planta Piloto, Laboratorio de Telecomunicaciones, entre otros. Si bien estos espacios son muy valiosos para las actividades previamente descritas, no cuentan en la actualidad con software de gestión que les permita administrar la oferta de servicios que potencialmente podrían soportar. El software que se propone en este trabajo podría no solo aportar soporte al Game Dev Lab sino también mediante el principio de CRS brindar soporte a estos otros laboratorios.

6.5 GLOSARIO

Treeshaking: Es un término comúnmente utilizado en el contexto de JavaScript para la eliminación de código muerto. WEBPACK. (S. F.).

Gafas VR: Conocidas más comúnmente como gafas de realidad aumentada, son unos dispositivos capaces de dar una experiencia de inmersión a los usuarios mediante una representación 3D. Pérez, JM. (2017, Marzo 8).

E-Sports: Son competiciones profesionales entre jugadores de algún videojuego. Las características para que un videojuego sea considerado e-sport son las siguientes. Casanova, J. (2017, abril 6).

- El juego permite el enfrentamiento directo entre dos o más participantes.
- Los jugadores compiten en igualdad de condiciones, siendo la victoria exclusivamente determinada por la habilidad de estos.
- Existen ligas y competiciones oficiales reguladas con reglas y formadas por equipos y jugadores profesionales.
- Goza de popularidad y de ser competitivo. Es decir, cuenta con miles de personas jugando y los medios de comunicación transmiten las competiciones.
- Debe promover el afán de superación.

Modelo de gestión: Los modelos de gestión de calidad señalan una serie de pautas para llevar a cabo una gestión eficaz. Los modelos de gestión y el enfoque basado en procesos. (2015, Octubre 7).

Aplicaciones distribuidas: Hace referencia al uso del modelo cliente servidor.

APIS: Una API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones. API significa interfaz de programación de aplicaciones. ¿Qué es una API? (s. f.).

6.6 MARCO TEÓRICO

6.6.1 (CRS)COMPUTER RESERVATION SYSTEM

El Computer Reservation System (CRS) es una arquitectura diseñada inicialmente por las aerolíneas, con el principal objetivo de cubrir sus propias necesidades de organización, posteriormente se extendió a las agencias de viajes como un canal y socio estratégico de incremento en las ventas. En este mismo marco, se desea aplicar la arquitectura de CRS como solución a la necesidad de un software de gestión de reservas para la Game Dev Lab y de igual manera dejar abierta la posibilidad a futuro de integrar otros laboratorios de la universidad (subsistemas).

Según Chang, P. (1992, October). Los CRS cuentan con 4 características principales:

1. Tienen un gran tamaño y su estructura es compleja, host con alta capacidad, redes amplias y heterogéneas. En conclusión debe tener una buena capacidad en cuestión de infraestructura.
2. Exige un alto rendimiento, debido a que este tipo de sistema debe soportar un número considerable de peticiones y de igual forma la cantidad de subsistemas puede ser amplia.
3. El estado de su operatividad es crítica ya que las entidades que hacen uso de los subsistemas dentro de la arquitectura dependen de él.
4. Los CRS comparten 3 componentes funcionales: servicios de información de reserva, redes de comunicación y aplicaciones locales de agente cliente (estas últimas son las que dan a cada entidad la posibilidad de administrar su propio subsistema).

En este sentido, se define que básicamente los Sistemas Centrales de Reservas – CRS, se crean como sistemas basados en el uso de las telecomunicaciones que son compartidos por diferentes competidores del sector y por empresas que mantienen relaciones con sus clientes, compradores o suministradores en la cadena de valor del negocio. No obstante Gómez Vieites, Á., & Suárez Rey, C. (2005), establecen que en un primer momento, Tecnologías de Información y Comunicación en el Sector Hotelero surgen los Sistemas Centrales de Reservas (CRS) que empiezan a vincular

a los diferentes agentes (proveedores e intermediarios) en los procesos de reservas hasta llegar a un concepto más amplio de sistemas de información denominados Sistemas Globales de Distribución (GDS) que integran todos los productos asociados a un cliente.

Como conclusión en el artículo de Buhalis, D. (2003). se precisa que el CRS representa un nivel inter organizativo; en donde las primeras aplicaciones de tecnologías de información y en su defecto de sistemas de información basados en redes, se desarrollaron con el fin de prestar apoyo a las empresas hoteleras y de transporte aéreo, constituyendo el antecedente previo de los actuales sistemas automatizados de reservas usados en la industria turística, pero, esto no quiere decir que no sea usado por los demás sectores de la industria tales como, cines, laboratorios ,salones, entre otros, los cuales gestionan la mayor parte de su información y servicio por medio de un Sistema de reservas computarizado.

6.6.2 (XML) Extensive Markup Language

XML (extensive Markup Language). Fue creado por el Word Wide Web Consortium (W3C). Su desarrollo comenzó en 1996 y la primera versión salió a la luz el 10 de febrero de 1998. La primera definición que apareció fue: Sistema para definir validar y compartir formatos de documentos en la web, García Arenas, M. I. (2005).

XML proporciona una particular forma de describir los datos estructurados. A diferencia de las etiquetas HTML, que se utilizan básicamente para controlar la presentación y apariencia de los datos, las etiquetas XML se utilizan para definir la estructura y los tipos de datos de los propios datos.

La implementación de la tecnología XML facilita el intercambio de información entre aplicaciones o sistemas heterogéneos para entornos Internet, intranet y extranet. Estos son algunos ejemplos donde se puede utilizar XML A. (2005).

- Un documento con etiquetas.
- Aplicaciones web de internet o intranet que mueven datos.
- Un registro de datos, como el conjunto de resultados de una consulta.

Además de tener amplias aplicaciones, XML presenta ciertas ventajas frente a los otros formatos a la hora de almacenar información, como lo son:

- XML se basa en texto, lo que lo hace más legible, fácil de documentar e incluso más fácil de depurar.
- XML puede utilizar una infraestructura ya creada para HTML, incluido el protocolo HTTP.
- XML tiene un análisis definido y ampliamente implementado, lo que hace posible la recuperación de información en diversos entornos.

6.6.3 SOFTWARE DE GESTIÓN

Un Software de Gestión es un sistema el cual está integrado por múltiples características y herramientas que se usan para ejecutar diversas tareas administrativas si se habla de estas individualmente, ya que cuando estas se ejecutan en conjunto los procesos operativos y productivos de una organización se simplifican.

Entre los softwares de gestión se destaca aquel que se integra a la perfección con las diferentes actividades de una empresa y dar el adecuado servicio a las necesidades que se puedan originar de las mismas. Generalmente se encargan de la gestión diaria y continua de diversos datos con los que interactúan los diferentes escenarios y procesos, los cuales se requieran cualquier día en determinada empresa, dependiendo de las necesidades presentadas, permitiendo así funciones como su inclusión, consulta, modificación, combinación, borrado, entre otras acciones, Programa Gestión Empresarial. (2016).

Unas de las ventajas más grandes que estos sistemas brindan son el control y la autonomía para los administradores, ya que al hacer uso de esta herramienta novedosa y digital pueden monitorear cualquier movimiento de la organización o empresa y de esta manera estar al tanto de las ganancias, inversiones, gastos y toda la información que sea convenientemente útil para la organización en determinado tiempo, Coral, M. A. B. (2019, mayo 14).

6.6.4 MODELO C/S-B/S

6.6.4.1 C/S Cliente servidor

C/S es una estructura de dos capas, cliente/servidor, el cual posee unas características muy importantes entre las cuales se destacan con gran notoriedad la fuerte interactividad, la seguridad de acceso, la velocidad de las respuestas rápidas, lo cual lo hace la mejor opción para el manejo de cantidades enormes de datos. Sin embargo, este tipo de estructura no tiene tanta flexibilidad para adaptarse a cambios, posee mala compatibilidad y un alto costo en el mantenimiento, sin dejar de lado la difícil gestión que este presenta, Yali, S., & Bisheng, F. (2015, November).

6.6.4.2 B/S Estructura de 3 capas

B/S es una estructura de tres capas, servidor de datos, servidor web y cliente, el cual destaca de entre sus principales características el fácil entendimiento del funcionamiento de su estructura posee buena flexibilidad, pero sin embargo presenta falencias en la seguridad, ya que tiene una seguridad bastante débil y una velocidad de transmisión lenta.

Por las razones anteriores este tipo de estructuras no son del todo viables cuando se trabajan individualmente en un proyecto, partiendo de las características del OLRS, lo ideal es combinar el C/S y B/S utilizando así los puntos fuertes de ambas estructuras dando como resultado un módulo orientado a múltiples usuarios con la

optimizada estructura de gestión del B/S y para la interactividad, seguridad e intercambio frecuente de datos haciendo uso de la sólida estructura de C/S, Ecu Red. (2020).

6.6.5 LABORATORIOS TRADICIONALES Y REMOTOS

Un laboratorio tradicional es un espacio físico donde se pueden realizar experimentos al cual pueden ingresar los estudiantes o personal perteneciente a la institución que lo administra. Su capacidad está definida por el tamaño del recinto y la cantidad de componentes disponibles. Por ende es susceptible a casos donde la cantidad de solicitudes es mayor a la capacidad del laboratorio. Debido a esto y al gran impacto de la tecnología en los últimos años es que las universidades recientemente han estado apostando por un nuevo tipo de laboratorio denominado laboratorio Remoto. Estos pueden llegar a tener una capacidad aún mayor y tener componentes adicionales que faciliten el desarrollo de las actividades para sus estudiantes.

Los Laboratorios Remotos (LR) son la representación del acceso a un laboratorio real, es decir el usuario accede a un laboratorio real con equipos reales, pero, de manera no presencial. Estos laboratorios están compuestos principalmente de un software y un hardware los cuales les permiten a los miembros de la comunidad educativa tener un acceso a prácticas como si estuvieran en un Laboratorio Tradicional. Zamora Musa, R. (2010).

Los Laboratorios Remotos generalmente requieren de una o más computadoras que sirvan de intermediarias entre el laboratorio y el control remoto externo del laboratorio, esta acción se efectúa gracias a un software específico que mediante una modalidad de streaming envía audio y video en tiempo de real del objeto bajo experimentación, de tal manera que el usuario que está realizando este proceso sienta la inmersión experimental como si la estuviese haciendo presencialmente.

A continuación, se listan las principales características y componentes de un Laboratorio Remoto funcional:

- Alumno, cliente o usuario: representa a la persona que va a efectuar el laboratorio remoto mediante una computadora.
- Servidor de enlace: es el que gestiona y administra las conexiones de los usuarios.
- Servidor multimedia: es el que provee al servidor de enlace las imágenes y el audio en tiempo real del ambiente experimental.
- Servidor de laboratorio: es el que da soporte a una aplicación en particular, en este caso, una placa de adquisición e instrumentación y está conectado al servidor de enlace.
- Objetivo experimental: objeto que está bajo el control del servidor de laboratorio y que permite adquirir o efectuar accionamientos, que se pueden medir, observar o escuchar gracias al servidor multimedia, Toderick, L., Mohammed, T., & Tabrizi, M. H. (2005, October).

6.7 ESTADO DEL ARTE

6.7.1 INTRODUCCIÓN

Durante la revisión de la literatura se analizó y concluyó que los **CRS (Computer Reservation System) o SRI (Sistemas de Reserva Informatizados)**, hoy en día están presentes en distintos campos de aplicación algo alejados para lo que el CRS fue diseñado principalmente. Pero aun así mantienen su esencia, Simplificar la reserva de elementos en un sistema en el cual la cantidad de usuarios es mayor a la cantidad de recursos disponibles, Wannous, M., Nakano, H., & Nagai, T. (2011, April). Como se mencionó anteriormente los **CRS** son usados en varios campos de aplicación como por ejemplo: bibliotecas, universidades, empresas de transporte entre otros. Cada uno con necesidades y soluciones diferentes. A continuación se mencionaran algunas de las metodologías y soluciones que fueron usadas para la elaboración de distintos tipos de **CRS**.

6.7.2 BADGER LAB

En la universidad de Stanford se ha desarrollado un software de gestión llamado Badger Lab, el cual mediante reglas de especificado XML; procesa los datos mediante parámetros establecidos para generar así las reservas. Lo que da como resultado una implementación eficiente y equitativa de los recursos del equipo, Murray, W., & Bell, M. (2012, July). Adicionalmente se podría decir que es un sistema flexible ya quees capaz de integrar y gestionar varios tipos de laboratorios. Los servicios que ofrecen: diversas estrategias para reservas de equipos, tarifas de equipos, datos de tiempo de ejecución y control de acceso de miembros.

6.7.3 SAHARA

El sistema del Game Dev Lab está enfocado para la reserva de equipos y elementos de un (LT) laboratorio tradicional, pero también existen sistemas de reserva para (LR) laboratorios remotos los cuales se encargan de gestionar el tiempo y recursos de los entornos virtuales destinados para la labor académica, su popularidad ha venido

creciendo recientemente debido a la necesidad de las universidades de tener entornos virtuales para la educación a distancia.

Un buen ejemplo podría ser la Universidad de East Carolina (ECU), la cual ha desarrollado un sistema de gestión que usa como sistema operativo Linux Red Hat y funciona mediante scripts bash de linux, esto se ejecutan de manera automática cada vez que un usuario inicia sesión y cuando finaliza la reserva. El hardware consta de varios elementos de equipos de control, como conmutadores de red y unidades de distribución de energía. Toderick, L., Mohammed, T., & Tabrizi, M. H. (2005, October). Este sistema consta con varios atributos que se consideran básicos para un funcionamiento idóneo.

- Implementación de un firewall que sólo permita el ingreso a personal autorizado y de igual forma que regule la salida de los datos autorizados por el laboratorio.
- Una Virtual Private Network (VPN), que encripte la actividad del usuario en redes no seguras como internet.
- Diseñar un sistema de reserva que permita a los usuarios hacer uso de la aplicación sin ninguna interferencia. también debe contar con reglas y restricciones para evitar la congestión o glitch a la hora de hacer un apartado de elementos en el LR
- Un diseño que permita a la universidad agregar o quitar fácilmente componentes de hardware de forma que no requiera modificaciones considerables.
- Los equipos deben ser capaces de restablecer su configuración a una condición predeterminada cuando la reserva haya expirado. Duffy, A. P., & Benson, T. M. (1992).

Adicionalmente se encontraron otras soluciones a sistemas de reserva para laboratorios remotos, que a pesar de tener varios puntos en común con la anterior también poseen elementos a tener en cuenta, como por ejemplo:

Según Schauer, F., Krbecek, M., Beno, P., Gerza, M., Palka, L., & Spilaková, P. (2015, February). REAMLABNET es un entorno virtual de laboratorio remoto. Este adicionalmente del sistema de reservas para recursos del laboratorio también tiene integrado varios módulos de simulación como por ejemplo péndulo simple, movimiento parabólico, movimiento rectilíneo uniforme entre otros. Otra característica adicional sería el almacenamiento de datos de los resultados obtenidos en cada sesión de laboratorio, estos datos son almacenados en un espacio de memoria previa asignado para cada integrante del laboratorio.

SAHARA Lowe, D. (2013). Es otra solución para los sistemas de reserva de laboratorios remotos. Su más notable característica es la implementación conjunta de los 2 modelos de reserva más comunes, sistemas basados en reservas y sistemas basados en colas. El primer modelo se encarga de asignar espacios de tiempo específicos para el uso de un recurso por otra parte el segundo acumula las tareas asignadas y las va ejecutando conforme el orden de llegada, sus datos son almacenados para que el investigador los pueda consultar tiempo después. Una solución bastante ingeniosa que permite disminuir el tráfico en el sistema y administrar de una forma eficiente los recursos.

Fig.N.1 Sistema Sahara

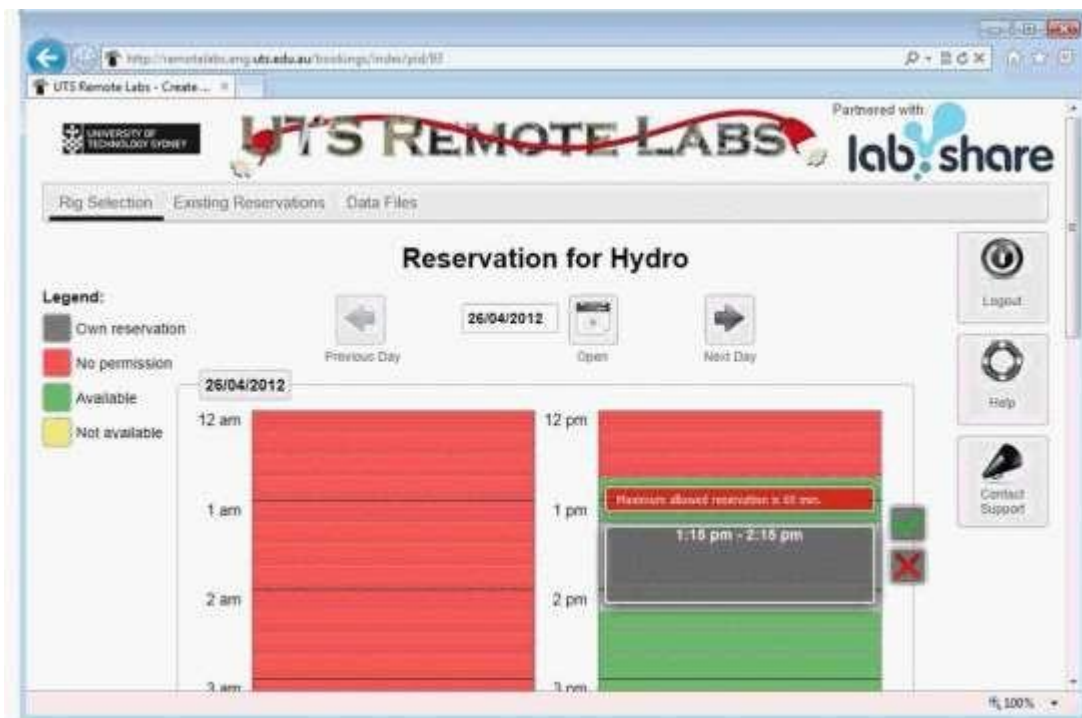


Fig.N.1 Dentro de Sahara, un usuario está haciendo una reserva para usar un equipo seleccionado Lowe, D. (2013).

6.7.4 C/S-B/S Y LABORATORIOS ABIERTOS

El **OLRS** (Opening Laboratory Reservation System) es un sistema de reservas para laboratorios abiertos apoyado por la universidad normal Yuxi de China. Esta solución implementa un modelo denominado C/SB/S que es la unión de 2 tipos de arquitecturas diferentes.

El C/S hace referencia a la arquitectura cliente servidor la cual ofrece una buena seguridad de acceso, velocidad de respuesta rápida, propicio para el manejo de grandes cantidades de datos; pero no es flexible tiene un alto costo y tiende a fallar si el tráfico es mayor a su capacidad. Yali, S., & Bisheng, F. (2015, November).

Por otra parte el B/S es una arquitectura de 3 capas presentación, negocios y acceso a datos, EcuRed. (S. F.). Sus características principales son: excelente flexibilidad, subdivisión de cargas de trabajo y fácil mantenimiento.; pero la velocidad de transmisión es lenta. Yali, S., & Bisheng, F. (2015, November).

Según lo implementado en la OLRs se puede deducir que el uso de estas 2 arquitecturas proporciona estabilidad al sistema, ya que si se requiere solo un intercambio de datos frecuente es usada la arquitectura C/S, como por ejemplo: solicitud de información de la página web, consulta de reservas o de recursos entre otros, y dado el caso si se requiere el análisis de datos para su posterior almacenamiento, como una reserva o la modificación del inventario es usada la arquitectura B/S. El uso de estas 2 arquitecturas reduce la carga de consultas mutuamente y optimiza la capacidad del sistema.

6.7.5 DESARROLLO DISPONIBLE A LA VENTA

Tecnoweb2, tecnoweb2. (S. F.). Es una empresa colombiana especializada en diseño y desarrollo de software. Creadora de un sistema de reservas para espacios y recursos que está disponible en el mercado actualmente. Fue desarrollado en PHP Y My SQL. Algunas de sus características son:

- Interfaz web amigable con el usuario y que puede ser accedido desde varios navegadores (Internet Explorer , Mozilla, Chrome)
- Genera informes respecto a la actividad que se presente en el sistema ya sean reservas o actualizaciones de inventario.
- Niveles de autenticación, según los permisos que tenga un usuario el sistema muestra diferentes interfaces un ejemplo básico: usuario y administrador.
- Reserva por tiempo o periodo, se puede hacer reservas en un lapso de tiempo corto como por ejemplo 2, 3 o 6 horas, o por periodo, días meses o años, según lo permitido por los administradores de los espacios o recursos.

- Multi idioma (traducidos al catalán, checo, chino, danés, holandés, finlandés, francés, alemán, griego, italiano, japonés, coreano, noruego, portugués, esloveno, español, sueco, turco).

6.7.6 SCHEDULING SYSTEM

Cada entidad que presta un servicio a una población que es mayor a sus recursos necesita un sistema de reserva, pero ¿es necesario crear un software de gestión desde 0 por cada tipo de entidad? En la Escuela de Graduados de Ciencia y Tecnología, Universidad de Kumamoto, (Japón). Creen que no, ya que existen herramientas web disponibles gratuitamente y que tienen un enorme potencial para este propósito. Wannous, M., Nakano, H., & Nagai, T. (2011, April).

Por esta razón crearon un sistema de programación (scheduling systems) que usa los servicios de google calendar para la gestión de un laboratorio remoto llamado NVLab. Su estructura es la siguiente:

Fig.N.2 Arquitectura NVLab

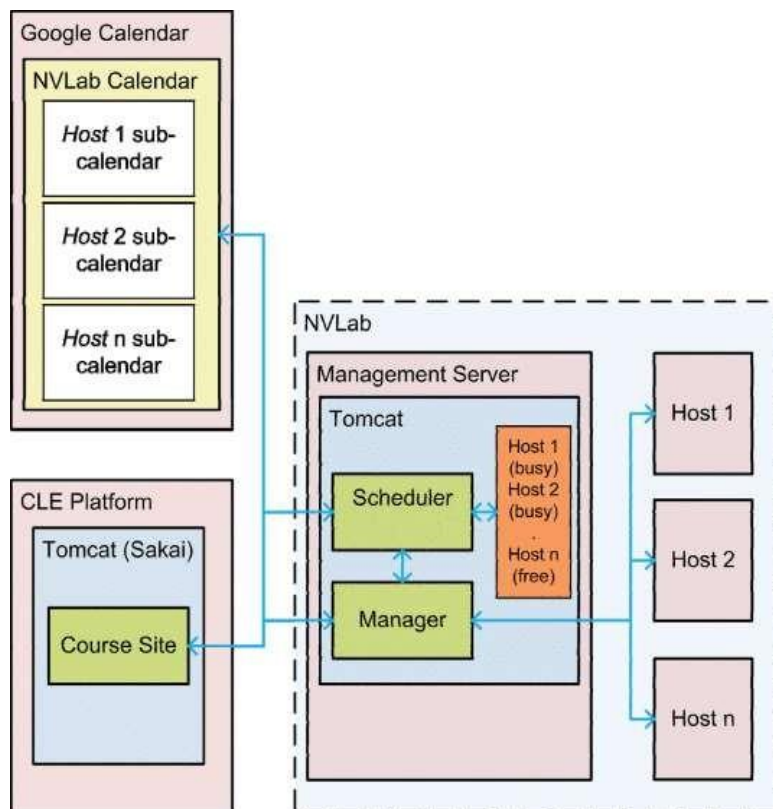


Fig.N.2 La arquitectura del sistema de programación en NVLab. Wannous, M., Nakano, H., & Nagai, T. (2011, April).

Desde una cuenta principal se crea un calendario el cual integra dentro de sí un itinerario propio para cada host. Cada reserva o ingreso a host se realiza desde una aplicación web que usa la DATA API de google calendar para mostrar la información, ya sea los calendarios, el formulario de reserva de host, reportes del sistema entre otros.

El sistema de reserva o uso de host es muy sencillo, el usuario entra y solicita un host, la aplicación revisa según los calendarios si hay algún host disponible, si es así le asigna un espacio, registra su ingreso y puede empezar a usar ese recurso de inmediato. Si no hay disponibilidad de un host se muestra al usuario el calendario de cada uno de los recursos para que el seleccione la hora adecuada para reservar el host. Una solución sencilla pero funcional a la hora de crear un sistema de reservas.

6.8 SELECCIÓN DE HERRAMINETAS

6.8.1 APLICACIÓN WEB

Según Luján-Mora, S. (2002). Una aplicación web es un conjunto de herramientas las cuales mediante un sistema de capas y la implementación de protocolos de comunicación como el HTTP, permite al usuario acceder a servicios o datos que se encuentran en servidores remotos. Este tipo de arquitectura se denomina cliente servidor y es usada para cualquier tipo de desarrollo web ya que permite crear aplicaciones distribuidas con un gran rendimiento.

Hoy en día, independientemente de si es una aplicación web, móvil, una aplicación de escritorio o un video juego, el trabajo de los desarrolladores se divide en 2 grupos, El frontend y el backend. El frontend es la capa externa que se comunica directamente con el cliente, recibe los datos y los prepara para su respectivo envío y procesamiento. Por otra parte, el backend contiene la lógica e interactúa con los recursos compartidos entre servidores, como por ejemplo conexión con APIS, bases de datos u otros servidores con lógica de negocio.

Para el desarrollo de cualquier aplicación sin importar su tamaño o tipo se necesita una estructura definida que ayude al programador a tener en orden su proyecto, un entorno que no solo le dé un estándar de desarrollo sino que también le permite automatizar procesos. Por esta razón es que entre la comunidad es habitual el uso de Frameworks, Los cuales ofrecen la estabilidad buscada.

A día de hoy existen varios Frameworks, cada uno de ellos enfocados en diferentes lenguajes de programación, tipos de desarrollo o estructuras. A continuación se mostraran los lenguajes y frameworks más usados hasta el año 2019.

Fig.N.3 Lenguajes más usados

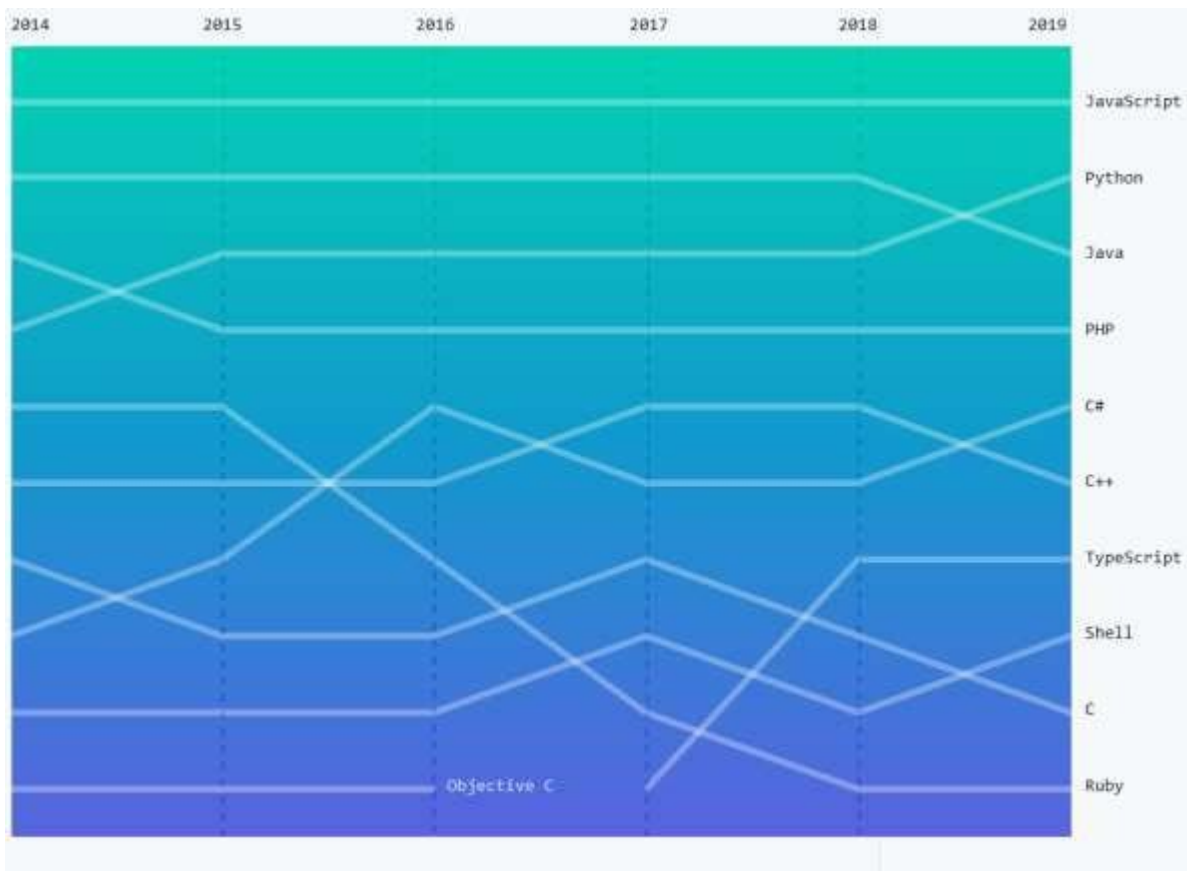


Fig.N.3 Lenguajes más usados desde el 2014 hasta el 2019. Según octoverse de github. GitHub(2020).

Fig.N.4 Frameworks más usados.

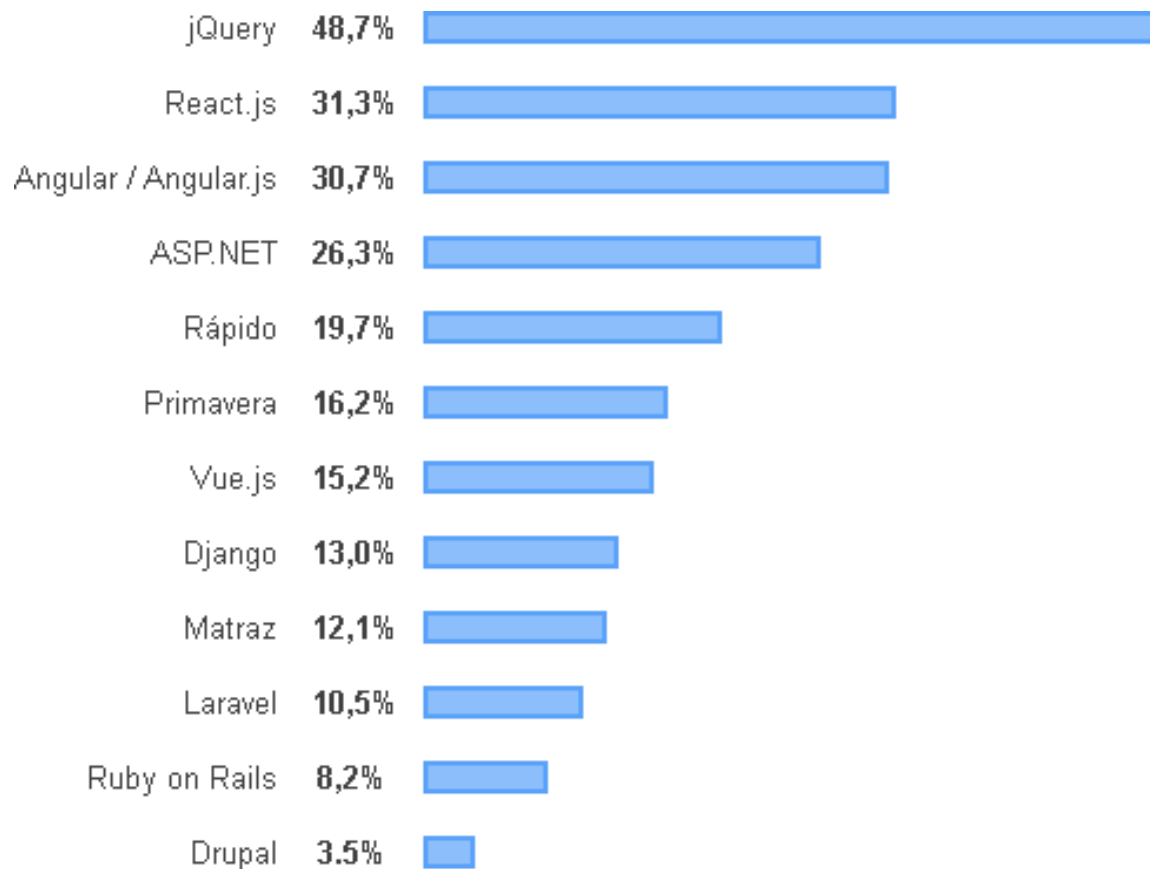


Fig.N.4 Frameworks más usados en el 2019 según stack overflow. Stack Overflow Developer Survey 2019. (2020)

Como se evidencia en la primer gráfica JavaScript es uno de los lenguajes más usados a la hora de programar, Esto no es una sorpresa ya que desde su lanzamiento en el año 1996 JavaScript mostró ser un lenguaje dinámico e innovador que poco a poco fue tomando terreno en el área del desarrollo web. En la segunda imagen hay un pequeño error ya que JQuery no es considerado un framework sino una librería de JavaScript y después de ella se encuentran los dos frameworks líderes del mercado React.js y Angular.

Si bien angular tiene una curva de aprendizaje un poco más empinada que React, angular es mucho más robusto permite mejor escalabilidad de proyectos y cuenta con una gran cantidad de componentes que facilitan el desarrollo. Esto se reafirma en la

última versión lanzada por el equipo de angular, que por cierto pertenece al gigante informático google.

En esta última Actualización (Angular 9), angular se ha enfocado en resolver varias problemáticas del pasado y darle un nuevo renacer implementando características nuevas como por ejemplo el uso de ivy como renderizador por defecto el cual mediante el tree shaking, elimina partes del código generado por angular que no se usan. Esto le da a los nuevos proyectos de angular, 2 características adicionales. El tiempo de su ejecución es entre un 40% y un 50% más rápida, y dependiendo del alcance del proyecto se podría obtener hasta un 40% de reducción de su tamaño. Fluin, S. (s. f.).

Fig.N.5 Tamaño de paquetes angular

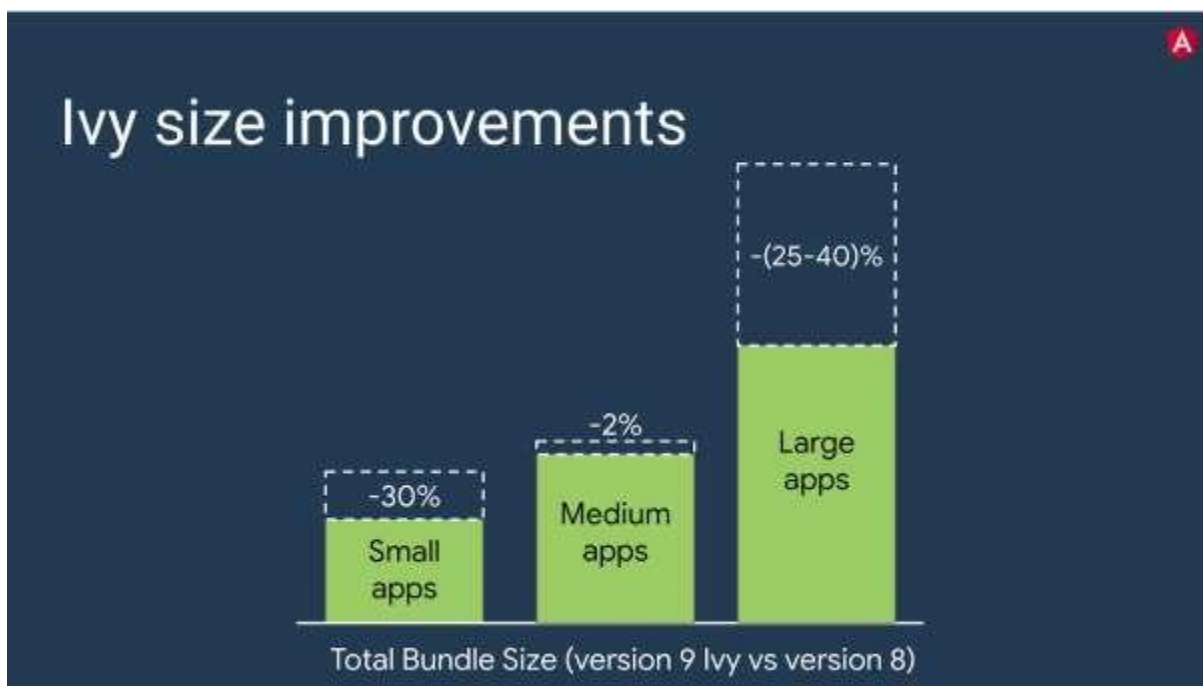


Fig.N.5 Tamaño de los paquetes Ivy versión 9 vs version9, Angular Versión 9. (2020).

Como es evidente, para escoger las herramientas de nuestro desarrollo hemos tenido en cuenta la popularidad de JavaScript en el mercado y la gran cantidad de componentes y soluciones ofrecidas por las recientes versiones de angular.

6.8.2 APLICACIÓN MÓVIL

Una aplicación móvil es un programa diseñado para ser ejecutado en una amplia gama de dispositivos móviles que se encuentran hoy en día en el mercado; las aplicaciones móviles permiten al usuario desarrollar cualquier tipo de actividad de una forma práctica. El desarrollo de aplicaciones móviles es reciente pero debido a su practicidad ha tomado bastante popularidad. Haciendo que instituciones educativas, empresas, prestadores de servicios entre otros recurren a su implementación. como por ejemplo la universidad Simón Bolívar de Barranquilla Colombia la cual ha desarrollado una aplicación móvil para la gestión de reservas en las salas de cómputo de su plantel educativo, Cassares, K. N., Ramos, B. O., Barrios, D. P., Cuadro, J. C. P., Lemus, L. R., & Martelo, E. (2015).

Las aplicaciones móviles se clasifican generalmente en tres tipos:

Fig.N.6 Tipos de aplicaciones móviles.



Fig.N.6 Tipos de aplicaciones móviles con sus diferentes características, Franklin, T. (2019, marzo 7).

- Aplicaciones nativas: Están diseñadas para sistemas operativos específicos con el fin de aprovechar al máximo los recursos del dispositivo, generalmente se descargan e instalan a través de una tienda de aplicaciones y se accede a ellas por medio de un ícono en el escritorio.
- Aplicaciones web o HTML5: Estas aplicaciones se basan en las tecnologías HTML5 las cuales son implementadas como un sitio web que muestra e

interactúa con el usuario ejecutándose en el navegador, dándole así al usuario la oportunidad de hacer uso de la app sin necesidad de haberla descargado e instalado.

- Aplicaciones híbridas: Como su nombre lo indica esas apps son una combinación entre apps nativas y web. Se obtienen generalmente de la tienda de aplicaciones y se aprovechan al máximo las características del dispositivo, De igual manera también se puede acceder a ellas desde un navegador; en conclusión, las apps híbridas se instalan como una app nativa y operan como una app web.

El desarrollo de aplicaciones móviles son aquellos procedimientos y procesos establecidos que interactúan cuando se crea un software para pequeños dispositivos inteligentes, del mismo modo, como se mencionó anteriormente en el apartado de aplicación web, las móviles tienen sus raíces en el desarrollo de software tradicional dividiendo el trabajo de desarrollo en dos partes: backend y frontend. Silva, F. (2019, diciembre 16).

Existen varios sistemas operativos actualmente en el mercado y hay diferentes herramientas tecnológicas que nos ayudan en el desarrollo de aplicaciones dependiendo de cada uno de los sistemas, a continuación se mostrarán los sistemas operativos Móviles más usados en la actualidad.

Fig.N.7 Sistemas Operativos más usados.

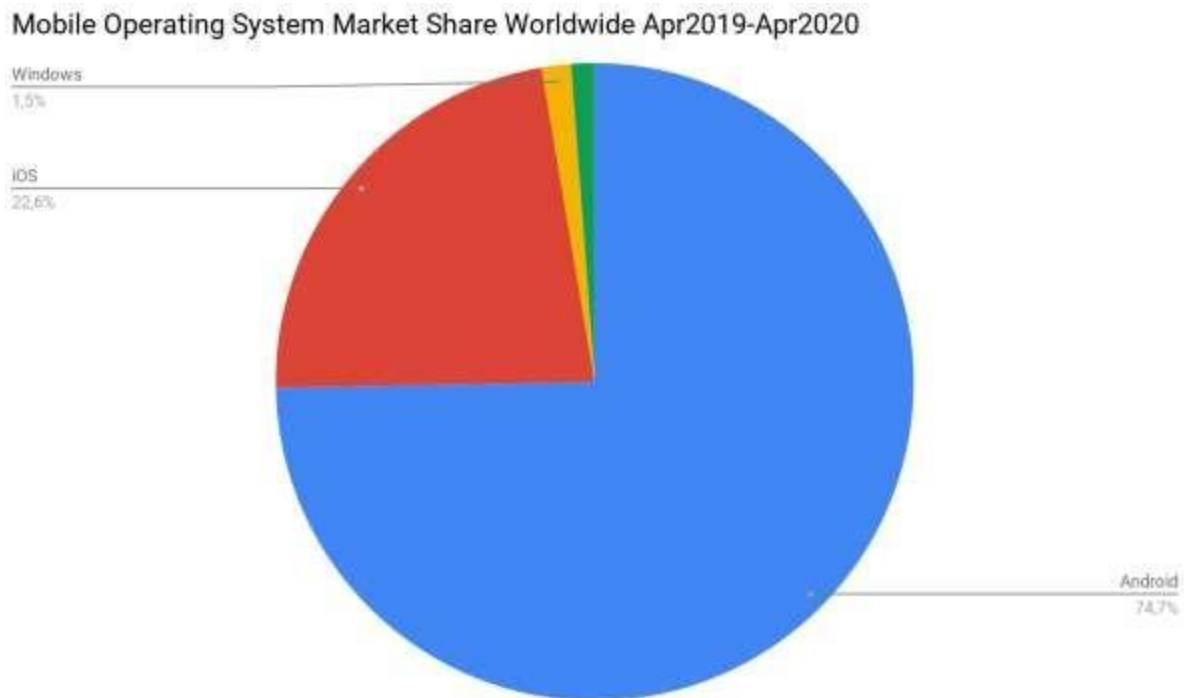


Fig.N.7 Sistemas operativos móviles más usados actualmente, Mobile Operating System Market Share Worldwide. (2020).

Como se puede apreciar en la gráfica el 80% de las personas tienen Android como sistema operativo en su teléfono, tan solo el 22,76% prefieren iOS y los demás sistemas operativos como KaiOS y Windows se encuentran por debajo del 1%, debido a esto elegimos realizar el desarrollo móvil en Android ya que tiene más aceptación del público y documentación de desarrollo y tecnologías.

Al día de hoy existen varios Frameworks, cada uno de ellos enfocados en diferentes lenguajes de programación, tipos de desarrollo o estructuras. A continuación, se mostrarán los lenguajes y Frameworks más usados para desarrollo móvil hasta el año 2019.

Fig.N.8 Lenguajes y Frameworks con mayor crecimiento.

01 Dart	532%
02 Rust	235%
03 HCL	213%
04 Kotlin	182%
05 TypeScript	161%
06 PowerShell	154%
07 Apex	154%
08 Python	151%
09 Assembly	149%
10 Go	147%

Fig.N.8 Lenguajes y Frameworks con mayor crecimiento e implementación en el periodo de 2018-2019, GitHub (2020).

Como se evidencia en la imagen Dart (Lenguaje de Flutter) es el lenguaje de programación móvil más usado en los últimos años. Flutter es un framework nuevo que optimiza la manera en la que se desarrollan las aplicaciones y para ello utiliza un lenguaje (Dart) que fue creado específicamente para este propósito. Flutter permite la creación de aplicaciones de una forma sencilla brindando la posibilidad de usar o crear widgets (son herramientas que nos permiten mostrar diferentes funcionalidades de la aplicación); todo esto lo hace, sin la necesidad de implementar ningún tipo de archivo JS o CSS para la funcionalidad y estilo de la aplicación, utilizando solamente un archivo .dart.

Inicialmente se pensó en usar NativeScript para el desarrollo de la aplicación móvil, sin embargo, al investigar más a fondo, en diferentes sitios oficiales de documentación de distintos frameworks se observó que además de NativeScript existen diversos Frameworks orientados al desarrollo móvil, específicamente Flutter, la razón principal por la cual se escoge Flutter para el desarrollo de la aplicación móvil ya que este framework al contrario de NativeScript no trabaja exponiendo las API'S nativas en JavaScript. Flutter trabaja con su propio lenguaje de programación (Dart), este lenguaje es compilado antes de tiempo o por adelantado en código, esto permite a la App comunicarse con la plataforma sin necesidad de cambiar el contexto a través de JavaScript, Flutter (2019).

6.9 RESULTADOS OBTENIDOS

La fase de Análisis se desarrolló durante el primer semestre del 2020 teniendo como resultado lo plasmado en los capítulos anteriores del presente documento. Donde se identificó el estado del arte, marco teórico y otros aspectos los cuales dieron más claridad con respecto a las tecnologías y como se están usando para dar solución a problemáticas similares a la que se planteó al inicio del proyecto. En esta fase se identificaron las necesidades del Game Dev Lab, según el tipo de servicios que son ofertados en el laboratorio, también se identificaron los actores del sistema y se definieron unas reglas generales para garantizar el correcto funcionamiento del sistema de reservas y el laboratorio.

Tabla 2. Descripción Actor Usuario.

<id>001	<USUARIO>
[Versión]	<02>(<30/05/2020>)
[Dependencias]	• <TECNICO LABORATORIO>
Descripción	<i>Este actor de negocio actual representa los usuarios que usaran el sistema para realizar las reservas de los diferentes elementos del laboratorio.</i>
Comentarios	<i>Según las reglas de negocio el usuario podrá ser sancionado por el TÉCNICO DEL LABORATORIO o por el sistema mismo. Y ser revocado su derecho a reserva o incluso el acceso a la plataforma.</i>

Tabla 3. Descripción actor Tecnico de Laboratorio.

<id>002	<TECNICO LABORATORIO>
[Versión]	<02>(<30/05/2020>)
[Dependencias]	• <USUARIO>
Descripción	<i>Este actor de negocio actual representa al personal encargado del laboratorio, el cual cumple funciones de administrador ya sea de los usuarios del sistema o los elementos del laboratorio.</i>

Comentarios

Otra de sus funciones es identificar posibles vulnerabilidades del sistema y reportarlas ante los desarrolladores.

Se identificaron las necesidades del laboratorio mediante charlas con el personal del mismo y del proyecto de investigación. De esta fase se obtuvo lo que tenemos como arquitectura del proyecto actualmente. Una arquitectura basada en servicios para poder implementar un sistema híbrido que funciones en los entornos web y móvil.

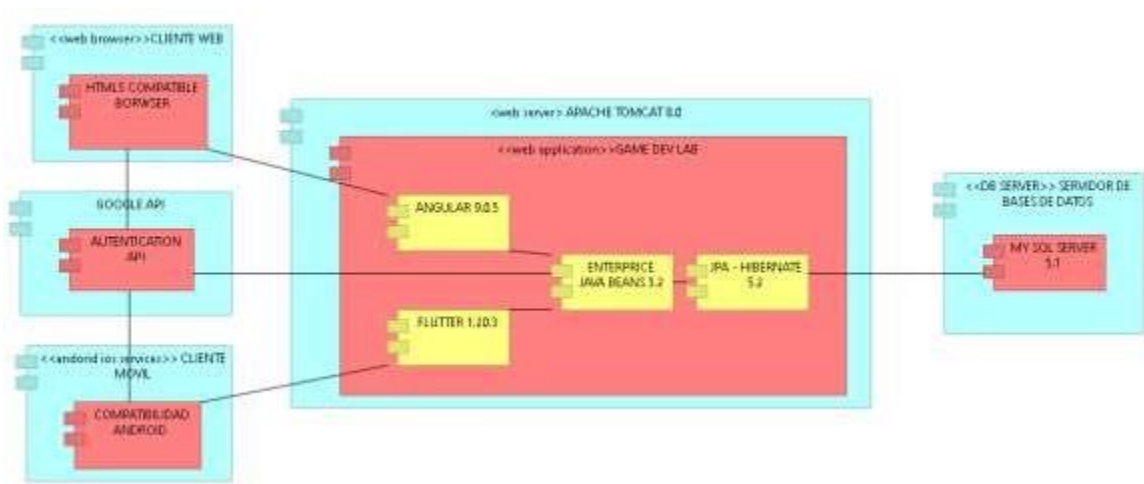


Figura 9. Arquitectura del sistema.

Los resultados de esta fase contribuyeron a la primera versión del siguiente anexo:

Anexo 1 (Documento de Análisis y Diseño).

En la fase de diseño se hicieron varios bocetos y "DUMMIES" que poco a poco fueron dando forma a lo que se quería ver en el Front del sistema de reservas, Los Cuales pueden ser vistos en el Anexo 2. Dummies.

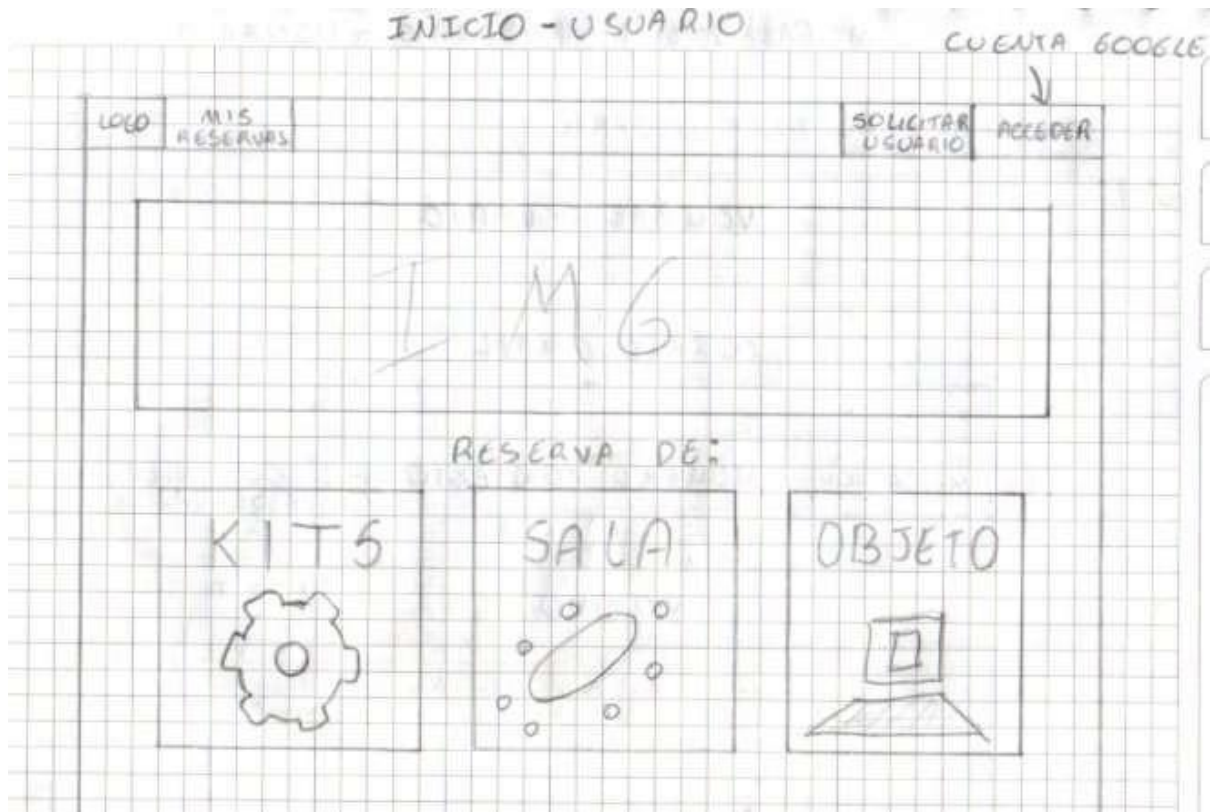


Figura 10. Dummie echo a mano.

Estos bocetos tenían en cuenta las necesidades del laboratorio encontradas durante la fase de análisis, de igual manera se tomó en cuenta los requisitos funcionales y la arquitectura planteada por el proyecto PAULO CESAR RAMIREZ PRADA, P. C. R. P., NITAE ANDRES URIBE ORDOÑEZ, N. A. U. O., & JUAN SEBASTIÁN CÁRDENAS ARENAS, J. S. C. A. (2017).

Durante esta fase se identificaron algunos problemas con la arquitectura planteada por el proyecto de investigación del laboratorio, por esta razón elaboramos una propuesta nueva la cual replanteo el paradigma planteado por los docentes, esta propuesta fue aceptada y el sistema pasó de ser basado en prioridades a un sistema de reservas clásico (las reservas de cada usuario son respetadas) el cual implementa aspectos nuevos como una ventana de reserva para los usuarios en general que les permita hacer sus solicitudes sin riesgo a que en cualquier momento llegase un usuario con mayor prioridad y lo quitase sus elementos. Para los usuarios con mayor prioridad o para la misma universidad esta restricción de la ventana de reservas no existe así son libres de reservar el espacio del laboratorio o elementos del mismo más allá de la ventana en cuestión.



Figura 11. Ventana de reserva.

En esta propuesta también se mantuvo El concepto de los tipos de elementos y reservas dentro del laboratorio ya que esto facilita la gestión de los mismos.

- Dos tipos de recursos: Fijos y variables, Los fijos serían elementos que solo pueden ser usados dentro del laboratorio, los variables no tendrían esta restricción. La naturaleza de los elementos es dada por el técnico de laboratorio y puede ser cambiada en cualquier momento.
 - FIJO (computadores sillas, mesas, televisor, entre otros)
 - VARIABLE (juegos de mesa, cascos, gafas vr , entre otros)
- Tipos de reserva
 - SALA (Requieren el espacio completamente para la actividad y ocupa todos los elementos fijos del laboratorio)
 - COMÚN (Requieren algunos de los elementos ya sean fijos o variables)

Para más detalle consultar el:

Anexo 3(Propuesta de Arquitectura Lógica).

Teniendo claro el camino que tomaría El sistema de reservas se elaboró una segunda versión del documento de Análisis y diseño en el cual se formalizaron aspectos Técnicos del proyecto como: Requisitos del sistema tales como:

- Login de usuario.
- Facilidad de Kits.

- Facilidad de reserva sala.
- Gestión de reservas.
- Registro de nuevos usuarios.
- Visualizar sanciones.
- Gestión de usuarios.
- Gestión de sanciones.
- Gestión de incidentes.
- Gestión de elementos.

De igual manera se identificaron los Casos de uso del sistema que sirvieron para especificar la comunicación y el comportamiento del sistema según su interacción con los usuarios y entre ellos mismos.

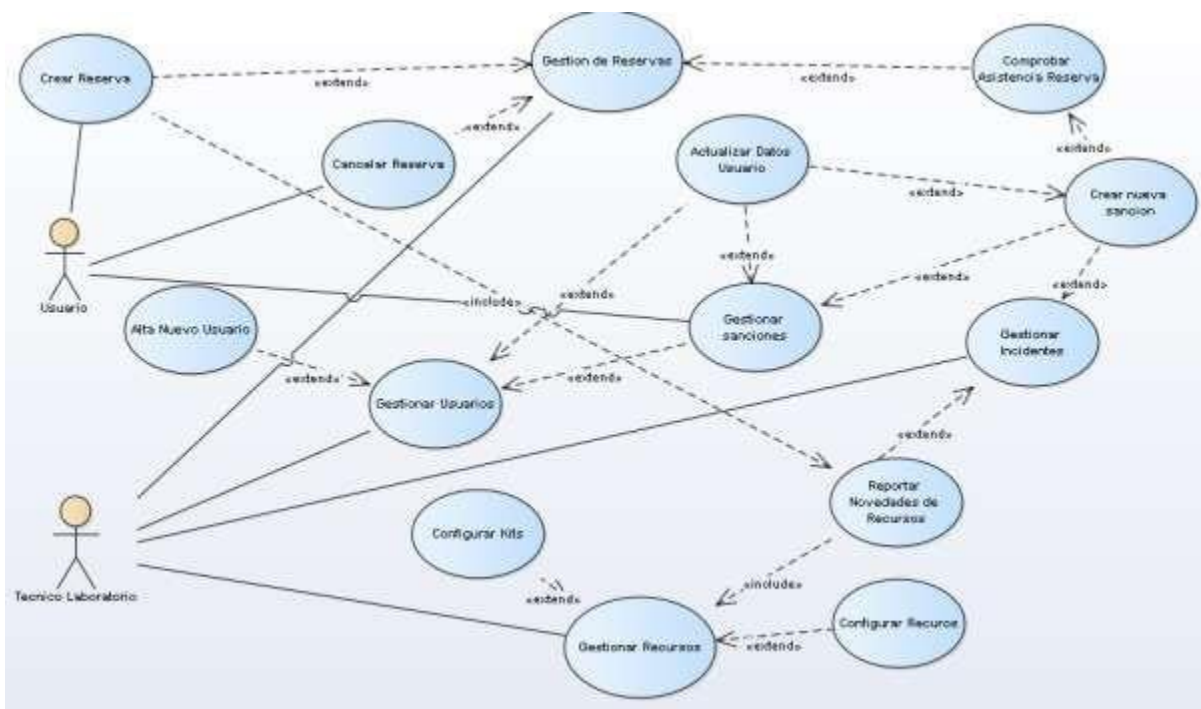


Figura 12. Diagrama de casos de uso.

Se creó un diagrama de clases la cual sirvió como base para que el proyecto de investigación del laboratorio creara la base de datos que sería usada en el back.

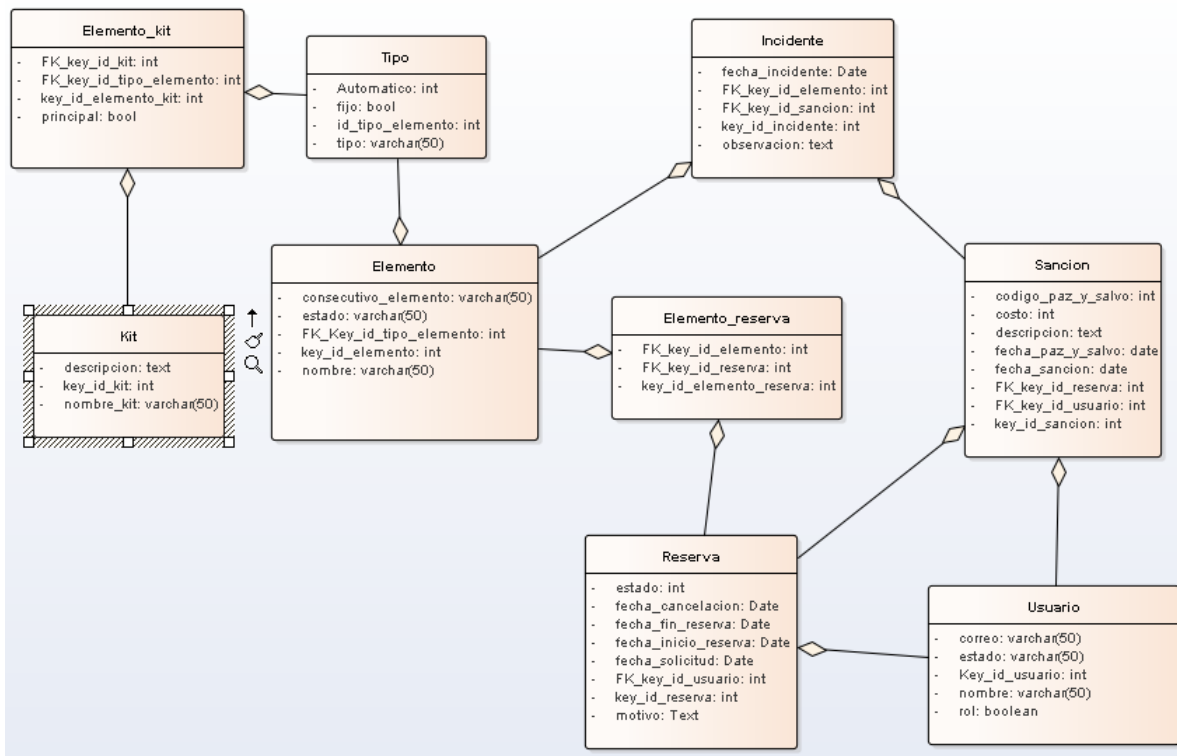


Figura 13. Diagrama de Clases.

Pueden consultar para más información en el:

Anexo 1 (Documento de Análisis y Diseño).

El desarrollo del sistema de reservas se llevó a cabo en 2 ciclos el primero fue la consolidación del Front, donde el punto de partida fueron los resultados obtenidos de las fases anteriores. Adicionalmente se implementaron librerías de diseño para mejorar el aspecto de la página WEB y la APP móvil para garantizar una buena aceptación por parte del público objetivo con respecto al Front. Tales como bootstrap, bootstrap watch y AOS.

Se usó la característica modular de angular para crear una organización de carpetas sencilla e intuitiva que facilitara la escalabilidad del proyecto.



Figura 14. Carpetas del Proyecto.

Se creó el servicio de autenticación usando angularx social login, de forma que pueda ser usado en cualquier lugar de la aplicación y verificar así en todo momento si el usuario esta logado y su respectivo rol.

```
TS auth-service.service.ts ×
src > app > services > TS auth-service.service.ts > AuthServiceService > loggedIn
1 | import { Injectable } from "@angular/core";
2 | import { SocialAuthService } from "angularx-social-login";
3 | import { FacebookLoginProvider, GoogleLoginProvider, } from "angularx-social-login";
4 | import { SocialUser } from "angularx-social-login";
5 | @Injectable({
6 |   providedIn: "root",
7 | })
8 | export class AuthServiceService {
9 |   user: SocialUser;
10 |   public loggedIn: boolean;
11 |   email: string;
12 |   logout: boolean;
13 |   constructor(private authService: SocialAuthService) {}
14 |   signInWithGoogle(): void {
15 |     this.authService.signIn(GoogleLoginProvider.PROVIDER_ID);
16 |   }
17 |   log() {
18 |     this.authService.authState.subscribe((user) => {
19 |       this.user = user;
20 |       this.loggedIn = user != null;
21 |       if (this.user) {
22 |         this.email = this.user.email;
23 |         if (this.email.includes("@unab.edu.co")) {
24 |           this.logout = true;
25 |         } else {
26 |           this.signOut();
27 |         }
28 |       }
29 |     });
30 |   }
31 |   getUser(): SocialUser {
32 |     return this.user;
33 |   }
34 |   signOut(): void {
35 |     this.authService.signOut();
36 |     if (this.logout) {
37 |       this.logout = false;
38 |     }
39 |   }
40 | }
```

Figura 15. Servicio angularx social login.

Para dejar el sistema listo para la implementación con el back se crearon los respectivos modelos de datos según la base de datos planteada por el proyecto de investigación.

En el segundo ciclo se trabajó en conjunto con el proyecto de investigación para la creación del API, Definiendo aspectos como la lógica del back, el desarrollo de la base de datos y la definición de los tipos de servicios que se modelarían.

El resultado de esta Fase es una arquitectura sólida que está lista para ser implementada en la práctica.



Figura 16. Inicio Sistema de Reservas.

Como evidencia, en esta fase se creó un manual técnico de usuario el cual muestra la interfaz del sistema y cómo sería su funcionamiento este documento puede ser consultado en el:

Anexo 4 (Manual Técnico de usuario WEB).

En la fase de pruebas, ya teniendo la estructura del sistema de reservas bien definido, se le presenta a un grupo de estudiantes y al técnico del laboratorio para obtener sus opiniones y sugerencias con respecto al correcto funcionamiento del sistema y a la aceptación del Front.

El resultado de estas pruebas de aceptación dio a entender que vamos por buen camino. La interfaz gráfica fue bastante aceptada. De igual manera la arquitectura lógica del sistema tuvo buenas críticas.

CALIFIQUE DEL 1 AL 5 LA INTERFAZ GRAFICA, SIENDO 1 EL MAS BAJO Y 5 EL MAS ALTO

15 respuestas

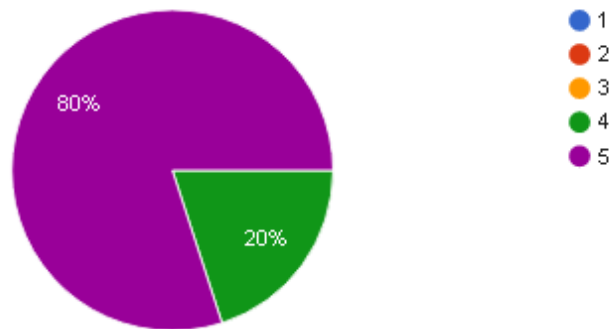


Figura 17. Pregunta de aceptación gráfica.

¿ES CLARA LA IDEA DE LOS KITS? ¿QUE SE ENTENDIO?

15 respuestas

Si muy clara. Es más fácil reservar el kit que incluye los implementos que se necesitan a reservar de a un elemento.

Si. Se entiende por ello que son pequeños paquetes para facilitar el préstamo de los mismos sin buscar uno a uno.

Sí, son reservas predeterminadas que sirven para agilizar el proceso.

Me parece una buena idea porque nos permiten encontrar y tener fácilmente diferentes cosas.

Sí, cuentan con lo requerido por la población objetiva de usuario.

Figura 18. Pregunta de aceptación lógica de los kits.

Para ver los Resultados de dichas pruebas de aceptación Consultar el:

Anexo 5 (Resultados Pruebas de Aceptación Web).

6.10 CONCLUSIONES

Este proyecto es solo el comienzo para dar paso a la implementación de un sistema que permita a los usuarios de la universidad gestionar las reservas del Game Dev Lab. Durante este proceso se identificaron los requerimientos según los servicios ofertados por el laboratorio y siguiendo las instrucciones de los técnicos y el proyecto de investigación del mismo. Identificando así la necesidad de implementar un CRS con arquitectura híbrida para que permitiese ser usada en entornos web y móvil. Esto dio las bases necesarias para dar paso a la fase de desarrollo, durante esta fase se modelaron los requerimientos obtenidos en las fases anteriores dando como resultado una interfaz gráfica web y móvil bastante aceptable. El desarrollo del front web en el framework angular fue una muy buena decisión este framework fue demasiado flexible con los requerimientos de la parte visual y lógica, gracias a que usa un lenguaje popular como lo es typescript el cual tiene demasiado soporte y documentación y a su estructura modular. Otra de las buenas características que Angular nos ofreció fue la creación de Servicios donde se pueden concentrar tareas según el usuario o funcionalidades., también se usó NPM como sistema de gestión de paquetes el cual permitió implementar librerías como ngx social la cual fue usada para la autenticación de los usuarios. Para finalizar se realizaron 2 pruebas de aceptación una para el personal del laboratorio (Técnicos de Laboratorio) y otra para un grupo de estudiantes, las 2 pruebas fueron bastante satisfactorias y se obtuvieron varias propuestas de mejora las cuales se tendrán en cuenta como trabajo a futuro. Para visualizar de forma más clara lo que se planea hacer con el proyecto a futuro pueden consultar el:

Anexo 6

(Plan de Trabajo).

		Implementación la base de datos			x	x														
2. Soporte	40 %	Definición protocolo de pruebas					x													
		Realización de pruebas y documentación					x	x												
		Análisis resultados y ajustes								x										
		Soporte al sistema instalado					x	x	x	x	x	x	x	x	x	x	x	x	x	x
3. Análisis y Diseño	10 %	Análisis de nuevos requerimientos	x	x	x	x														
		Diseño de nueva versión del sistema				x	x	x	x											
		Diseño de nuevas interfaces (móvil, web)				x	x	x	x											
4. Desarrollo	20 %	Configuración repositorio y control de versión	x	x																
		Desarrollo de nuevas funcionalidades						x	x	x	x	x	x	x						
		Actualización manual técnico y de usuario											x	x						

7 REFERENCIAS BIBLIOGRÁFICAS

- PAULO CESAR RAMIREZ PRADA, P. C. R. P., NITAE ANDRES URIBE ORDOÑEZ, N. A. U. O., & JUAN SEBASTIÁN CÁRDENAS ARENAS, J. S. C. A. (2017). Consolidación del Game Dev Lab mediante un modelo y un software de gestión sostenible, que permitan ofrecer un portafolio de servicios que integren los sectores empresarial, académico y comunidad gamer... Recuperado 8 febrero, 2020, de <https://docs.google.com/document/d/1X4cH1sw2NN9II1BqGX7SfGQsR5p8Vd8jwOn07sgMYEU/edit?usp=sharing>
- Luján-Mora, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. Editorial Club Universitario.
- Velandia, A. T., & Carlos, V. O. J. (2006). Desarrollo de un Aplicativo Web para la Reserva de Salas de Informática, Multimedia y Equipos Audiovisuales en la Universidad de San Buenaventura Sede Bogotá (Doctoral dissertation, Tesis de grado Tecnólogo en Sistemas, Bogotá, Colombia: Universidad San Buenaventura de Bogotá).
- Matamoros Motoche, M. A., & Sánchez Idrovo, M. F. (2011). Análisis, diseño y creación de una aplicación web para el control de préstamos y consultas en la Biblioteca de la Municipalidad del cantón Biblián (Bachelor's thesis).
- Cassares, K. N., Ramos, B. O., Barrios, D. P., Cuadro, J. C. P., Lemus, L. R., & Martelo, E. (2015). Sistema de reserva de salas de cómputo en la Universidad Simón Bolívar-Reserva Lab. Investigación y desarrollo en TIC, 6(2), 68-75.
- UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA. (2019, marzo 6). ACUERDO N 103. Recuperado 2 de marzo de 2020, de <https://miportalu.unab.edu.co/Archivos/?p=calidad/Acuerdo-103.pdf>
- WEBPACK. (s. f.). Tree Shaking. Recuperado 6 de marzo de 2020, de <https://webpack.js.org/guides/tree-shaking/>
- Perez, JM. (2017, marzo 8). Qué son y para qué sirven las gafas VR. Recuperado 28 de abril de 2020, de <https://www.hijosdigitales.es/es/2017/02/sirven-las-gafas-vr/>

- Casanova, J. (2017, abril 6). ¿Qué son los deportes electrónicos? Recuperado 28 de abril de 2020, de https://as.com/esports/2016/09/16/portada/1473981525_248613.html
- Los modelos de gestión y el enfoque basado en procesos. (2015, octubre 7). Recuperado 28 de abril de 2020, de <https://www.isotools.org/2015/03/03/los-modelos-de-gestion-y-el-enfoque-basado-en-procesos/>
- ¿Qué es una API? (s. f.). Recuperado 28 de abril de 2020, de <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>
- Chang, P. (1992, October). Computerized reservation systems. In [Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics (pp. 1247-1249). IEEE.
- Gómez Vieites, Á., & Suárez Rey, C. (2005). Sistemas de información: herramientas prácticas para la gestión empresarial. *Madrid: Ra-Ma Editorial*.
- Buhalis, D. (2003). eTourism: Information technology for strategic tourism management. Pearson education.
- A. (2005). Technical documentation, API, and code examples. Recuperado 2020, de <https://docs.microsoft.com/en-us/?url=/library/SPA/vbcon/html/vbconXMLTechnologyBackgrounder.asp>.
- García Arenas, M. I. (2005). Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada. Recuperado 2020, de <http://geneura.ugr.es/~maribel/xml/introduccion/index.shtml>.
- Programa Gestión Empresarial. (2016). Programa Gestión Empresarial. Recuperado 2020, de <https://www.programagestionempresarial.com/que-es-un-software-de-gestion/>
- Coral, M. A. B. (2019, mayo 14). ¿Qué es un software de gestión? Recuperado 2020, de <https://www.siigo.com/blog/empresario/software-de-gestion/>
- Yali, S., & Bisheng, F. (2015, November). Research and Design on Opening Laboratory Reservation System. In 2015 7th International Conference on Information Technology in Medicine and Education (ITME) (pp. 533-536). IEEE.
- EcuRed. (2020). Arquitectura de tres niveles - EcuRed. Recuperado de https://www.ecured.cu/Arquitectura_de_tres_niveles
- Zamora Musa, R. (2010). Laboratorios remotos: análisis; características y su desarrollo como alternativa a la práctica en la Facultad de Ingeniería.

- Toderick, L., Mohammed, T., & Tabrizi, M. H. (2005, October). A reservation and equipment management system for secure hands-on remote labs for information technology students. In *Proceedings Frontiers in Education 35th Annual Conference* (pp. S3F-S3F). IEEE.
- Silva, F. (2019, diciembre 16). Definición y cómo funcionan las aplicaciones móviles. Recuperado de <https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>
- Flutter - Crea hermosas aplicaciones nativas en tiempo récord. (2019). Recuperado 2020, de <https://flutter-es.io/>
- Wannous, M., Nakano, H., & Nagai, T. (2011, April). Google Calendar™ for managing and monitoring the utilization of a web-based laboratory's resources. In *2011 IEEE Global Engineering Education Conference (EDUCON)* (pp. 210-213). IEEE.
- Murray, W., & Bell, M. (2012, July). Badger Lab Management Software. In *2012 19th Biennial University/Government/Industry, Micro/Nano Symposium (UGIM)* (pp. 1-1). IEEE.
- Toderick, L., Mohammed, T., & Tabrizi, M. H. (2005, October). A reservation and equipment management system for secure hands-on remote labs for information technology students. In *Proceedings Frontiers in Education 35th Annual Conference* (pp. S3F-S3F). IEEE.
- Duffy, A. P., & Benson, T. M. (1992). Resonant applicator design for industrial microwave heating. *Electronics Letters*, 28(20), 1878-1879.
- Schauer, F., Krbecek, M., Beno, P., Gerza, M., Palka, L., & Spilaková, P. (2015, February). REMLABNET II-open remote laboratory management system for university and secondary schools research based teaching. In *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)* (pp. 109-112). IEEE.
- Lowe, D. (2013). Integrating reservations and queuing in remote laboratory scheduling. *IEEE Transactions on Learning Technologies*, 6(1), 73-84.
- Yali, S., & Bisheng, F. (2015, November). Research and Design on Opening Laboratory Reservation System. In *2015 7th International Conference on Information Technology in Medicine and Education (ITME)* (pp. 533-536). IEEE.
- EcuRed. (s. f.). Arquitectura de tres niveles - EcuRed. Recuperado 28 de abril de 2020, de https://www.ecured.cu/Arquitectura_de_tres_niveles

- tecnoweb2. (s. f.). Software Web Para Reserva de Espacios / Recursos. Recuperado 2 de marzo de 2020, de <https://tecnoweb2.com/software-reserva-aulas-salas-audiovisuales-colombia>
- Fluin, S. (s. f.). Version 9 of Angular Now Available — Project Ivy has arrived! Recuperado 2 de marzo de 2020, de <https://blog.angular.io/version-9-of-angular-now-available-project-ivy-has-arrived-23c97b63cfa3>
- Stack Overflow Developer Survey 2019. (2020). Recuperado de <https://insights.stackoverflow.com/survey/2019#developer-profile>
- GitHub (2020). Recuperado de <https://octoverse.github.com/#trends>
- Angular Version 9. (2020). Recuperado de <https://blog.angular.io/version-9-of-angular-now-available-project-ivy-has-arrived-23c97b63cfa3>
- Franklin, T. (2019, Marzo 7). Mobile App Testing Challenges Rise with Higher User Expectations. Recuperado de <https://www.qualitylogic.com/2018/06/11/mobile-app-testing-challenges-rise-with-higher-user-expectations/>
- Mobile Operating System Market Share Worldwide. (2020). Recuperado de <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201904-202004+>