



Mendoza, Matías Joaquín

Sistema de documentación de incidentes relacionados a diferentes tecnologías que administra la organización

2019

Instituto de Ingeniería y Agronomía Ingeniería en Informática



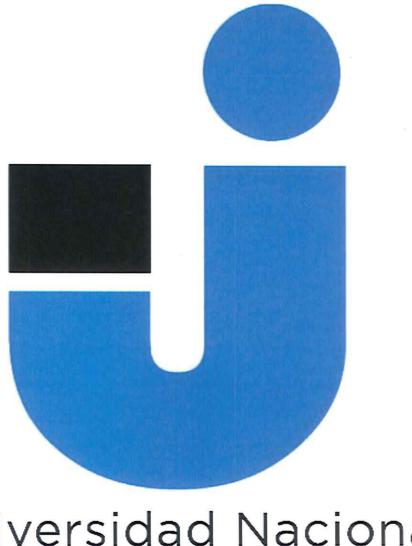
Esta obra está bajo una Licencia Creative Commons Internacional. Atribución - No Comercial - Compartir Igual 4.0 https://creativecommons.org/licenses/by-nc-nd/4.0/

:umento descargado de RID - UNAJ Repositorio Institucional Digital de la Universidad Nacional Arturo Jauretche

Cita recomendada:

Mendoza, M. J. (2019). Sistema de documentación de incidentes relacionados a diferentes tecnologías que administra la organización [Informe final]. Universidad Nacional Arturo Jauretche.

Disponible en RID - UNAJ Repositorio Institucional Digital UNAJ https://biblioteca.unaj.edu.ar/rid-unaj-repositorio-institucional-digital-unaj



Universidad Nacional ARTURO JAURETCHE

Ingeniería en informática **Práctica profesional supervisada**Informe final
Matías Joaquín Mendoza



Práctica Profesional Supervisada (PPS)

Página 2 de 72

PRÁCTICA PROFESIONAL SUPERVISADA (PPS) Título de la PPS Informe de Avance

DATOS DEL ESTUDIANTE

Apellido y Nombres: Mendoza, Matías Joaquín

DNI: 38.439.920 Nº de Legajo: 8852

Correo electrónico: matiasmendoza1994@gmail.com

Cantidad de materias aprobadas al comienzo de la PPS: 45

PPS enmarcada en artículo 7 inciso "b". de la Resolución (CS) 103/16.

DOCENTE SUPERVISOR

Apellido y Nombres: Dr. Conde, Sergio Daniel Correo electrónico: drcondesergio@gmail.com

DOCENTE TUTOR DEL TALLER DE APOYO A LA PRODUCCIÓN DE TEXTOS ACADÉMICOS DE LA UNAJ

Apellido y Nombres: Prof. Medina, Patricia Correo electrónico: medinapatf@gmail.com

DATOS DE LA ORGANIZACIÓN DONDE SE REALIZA LA PPS

Nombre o Razón Social: Novared S.A.

Dirección: Maipú 464 piso 5°

Teléfono: 4000-0300

Sector: Seguridad Informática

TUTOR DE LA ORGANIZACIONAL

Apellido y Nombres: Ing. Lopez, Jorge Correo electrónico: jalopez@novared.net

FIRMA DEL COORDINADOR DE LA CARRERA

Firma Estudiante:

Firma Docente
Supervisor:

Firma docente tutor
TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 3 de 72

Índice

Índice	3
Cronograma de trabajo	4
Resumen	6
Objetivos	8
Tareas a ejecutar	9
Desarrollo	12
Relevamiento y análisis funcional	12
Metodología de trabajo	14
Requisitos del sistema	16
Requisitos Funcionales	17
Requisitos No Funcionales	17
Arquitectura y tecnologías a utilizar	18
Arquitectura actual	18
Propuesta de Arquitectura a implementar	19
Tecnologías a utilizar	21
Desarrollo del sistema solicitado	35
Pruebas y ajustes del sistema solicitado	62
Conclusión	64
Reflexión sobre la Práctica Profesional Supervisada como espacio de forr	nación 65
Bibliografía	67
Anexos	68

Firma Estudiante:	Firma Docente,	Firma docente tutor	Firma tutor
	Supervisor:	TAPTA:	Organizacional:
Zam.	4	Maris	1/1/2-
	Dr. SERGIO DANIEL CONDE	(My	A THE WAY
- 1000	DOCTOR EN PSICOLOGIA SOCIAL	9	7//



Práctica Profesional Supervisada (PPS)

Página 4 de 72

Cronograma de trabajo

Hitos	Descripción de Actividades	Tiempos Estimados	Estado actual
Base d	e Conocimiento		
	Relevamiento y Análisis funcional del sistema solicitado Recolectar requerimientos con los interesados del proyecto Analizar la arquitectura disponible actualmente Definir la arquitectura de la aplicación Analizar y definir las tecnologías a utilizar		Realizada
Etapa 1	Confección del documento con la definición técnica-funcional	120 Hs	Realizada
	 Desarrollo del sistema solicitado Desarrollar la base de datos dentro del sistema de gestión Desarrollar un módulo de integración con el sistema de gestión Desarrollar la aplicación web 		Realizada
Etapa 2	Código fuente desarrollado	180 Hs	Realizada

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor TAPTA:

Firma docente tutor TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 5 de 72

Hitos	Descripción de Actividades	Tiempos Estimados	Estado actual	
	Base de Conocimiento			
	Pruebas y ajustes del sistema solicitado Realizar pruebas y ajustes sobre el módulo de integración con el sistema de gestión Realizar pruebas y ajustes sobre la aplicación web Realizar pruebas y ajustes integrales del sistema		Pendiente	
Etapa 3	Código fuente testeado integralmente	120 Hs	Pendiente	
	Despliegue del sistema en un ambiente productivo Implementar la aplicación web en el ambiente productivo Documentar la utilización del sistema como usuario final Realizar un Skill-Transfer a los usuarios de la aplicación Evaluación del sistema por parte de los usuarios Realización de ajustes post producción		Pendiente	
Etapa 4	Sistema funcionando de forma productiva	60 Hs	Pendiente	

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor TAPTA:



Práctica Profesional Supervisada (PPS)

Página 6 de 72

Resumen

En este artículo se presentará el diseño, desarrollo y la implementación de una aplicación web destinada a la documentación de incidentes relacionados a las diferentes tecnologías que administra la organización.

Este sistema será implementado en Novared S.A. una empresa de servicios profesionales en seguridad informática como Base de Conocimiento con el objetivo de ser utilizado para la búsqueda de soluciones en problemas recurrentes que ya han sido documentados.

El beneficio final de la aplicación es el de agilizar la resolución de problemas por parte de los técnicos que podrán acceder de forma rápida y sencilla a los artículos para la resolución de sus incidentes, evitando así tener que replicar el troubleshooting¹ ya realizado para el inconveniente a tratar.

Dicha aplicación consiste en un sitio web con las funcionalidades de creación, edición y eliminación de artículos dentro de la base de conocimiento, la búsqueda mediante la utilización de distintos filtros que facilitan el acceso rápido a la información necesaria, la exportación en un formato de documento portable (pdf) y la conexión con el sistema de gestión empresarial que se utiliza en la compañía.

Los usuarios que tendrán acceso serán tanto los técnicos y analistas de la empresa como también sus clientes. Estos podrán acceder al portal para consultar de manera online diferentes alternativas de solución a sus problemas. En adición, en el caso de no encontrar el incidente, se podrá generar un ticket de soporte en el sistema de gestión de tickets ofrecido por la firma.

Firma Estudiante:

Firma Docente Supervisor:

> Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor

TAPTA:

¹ Troubleshooting se entiende por la búsqueda de los problemas que ocurren con el fin de aplicar una solución a estos. Esta frase es comúnmente utilizada en ámbitos informáticos.



Práctica Profesional Supervisada (PPS)

Página 7 de 72

Una vez implementado el sistema final, se contempla un tiempo de prueba por parte de los usuarios en el cual se podrán reportar errores y posibles mejoras al equipo de desarrollo con el fin adaptar el sistema para una mayor usabilidad.

Firma Estudiante:

Firma Docente/ Supervisor: Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL

Firma docente tutor TAPTA:



Práctica Profesional Supervisada (PPS)

Página 8 de 72

Objetivos

El objetivo del presente trabajo es desarrollar un sistema productivo a utilizar como "Base de Conocimiento" en el cual se consolide toda la información relevante para la resolución de los problemas reportados por los técnicos sobre las diferentes tecnologías administradas por la empresa.

A continuación, se listan los objetivos específicos que deben concretarse para la conclusión del proyecto:

- Analizar problemas de la metodología anterior
- Buscar alternativas de solución a los problemas encontrados
- Disponer de repositorio histórico de incidentes
- Permitir la correcta carga de datos relevantes
- Diseñar una estructura organizada para la búsqueda por categorías
- Poner a prueba la aplicación para realizar los ajustes necesarios
- Sistematizar el uso de la aplicación para:
 - o Optimizar los tiempos de resolución de problemas
 - Evitar la duplicidad de tareas
 - Optimizar los tiempos de respuesta al cliente
 - Motivar a los usuarios a la carga de incidentes

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE

Firma docente tutor

TAPTA:



Práctica Profesional Supervisada (PPS)

Página 9 de 72

Tareas a ejecutar

Para el cumplimiento del proyecto se han planificado una serie de etapas que engloban un conjunto de tareas relacionadas entre sí. Dichas etapas se listan a continuación:

Etapa 1: Relevamiento y Análisis funcional del sistema solicitado
 En esta primera etapa se busca un entendimiento general del sistema necesitado, de modo que se permita la confección de un documento que consolide la definición completa de la aplicación a desarrollar.

Las tareas que se estiman para esta etapa son las siguientes:

- Recolectar requerimientos con los interesados del proyecto
- Analizar la arquitectura actual
- Definir la arquitectura de la aplicación
- Analizar y definir las tecnologías a utilizar
- Hito 1:

Entregable 1 "Definición técnica-funcional del sistema requerido"

Firma Estudiante:

Firma Docente/ Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor



Práctica Profesional Supervisada (PPS)

Página 10 de 72

• Etapa 2: Desarrollo del sistema solicitado

En esta segunda etapa se da comienzo al desarrollo de la aplicación requerida con base a lo definido en el documento entregable de la primera etapa.

Las tareas que se estiman para esta etapa son las siguientes:

- Desarrollar la base de datos dentro del sistema de gestión
- Desarrollar un módulo de integración con el sistema de gestión
- Desarrollar la aplicación web
- Hito 2: Entregable 2 "Código fuente del sistema desarrollado"

Etapa 3: Pruebas y ajustes del sistema solicitado

En esta tercera etapa se realizan las pruebas sobre el código fuente desarrollado en la etapa anterior. En adición se implementan los ajustes pertinentes para un correcto funcionamiento del sistema (en el caso que sea necesario).

Las tareas que se estiman para esta etapa son las siguientes:

- Realizar pruebas y ajustes sobre el módulo de integración con el sistema de gestión
- Realizar pruebas y ajustes sobre la aplicación web
- Realizar pruebas y ajustes integrales del sistema
- Hito 3: Entregable 3 "Código fuente testeado en forma unitaria e integral"

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
	Dr. SERGIO DANIEL CO	OKIAL (M)	1/A



Práctica Profesional Supervisada (PPS)

Página 11 de 72

Etapa 4: Despliegue del sistema en un ambiente productivo
 En esta última etapa se implementa, dentro de un ambiente productivo, la aplicación web probada en la etapa anterior. En adición, se realiza la confección de un documento para usuarios finales y una capacitación a estos últimos.

Las tareas que se estiman para esta etapa son las siguientes:

- Implementar la aplicación web en el ambiente productivo
- Documentar la utilización del sistema como usuario final
- Realizar un Skill-Transfer a los usuarios de la aplicación
- Evaluación del sistema por parte de los usuarios
- Realización de ajustes post producción
- Hito 4: Sistema implementado en el ambiente productivo de la compañía

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 12 de 72

Desarrollo

Relevamiento y análisis funcional

La primera tarea a ejecutar en el transcurso del proyecto es "Recolectar requerimientos con los interesados del proyecto" correspondiente a la etapa 1 del cronograma de trabajo.

Esta tarea busca consolidar toda la información respecto al problema planteado por la empresa, realizar un análisis de esta información y, sobre la base de este análisis, listar los requisitos de software que deben cumplirse para desarrollar un sistema que solucione el inconveniente propuesto.

El problema manifestado es la ineficiencia en la administración de tiempos por parte de los técnicos y analistas de la firma al momento de la resolución de tareas. Este percance se debe a la incorrecta comunicación entre los diferentes equipos que trabajan en la compañía, lo que genera duplicidad de tareas a la hora de investigar posibles soluciones a los inconvenientes presentados.

En el sistema actual, al detectar un inconveniente, un técnico se encarga de verificar que el caso tenga solución, utilizando sus conocimientos o haciendo consultas a sus colegas y en el caso de no conocerla, se realiza una investigación para dar con la forma de solventar este asunto.

Existen ocasiones en las que un problema reportado ya ha ocurrido con anterioridad, por lo que una manera sencilla de aplicar las resoluciones seria recordar el procedimiento realizado, aunque no siempre se cuenta con la persona encargada de aplicar la solución en su momento, lo que lleva a la realización de un trabajo hecho anteriormente, invirtiendo de nuevo el tiempo en la investigación de

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE

Firma docente tutor

TAPTA:

Firma tutor

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 13 de 72

las posibles soluciones. El flujo de trabajo que se sigue en el proceso actual se puede notar en la Figura 1.

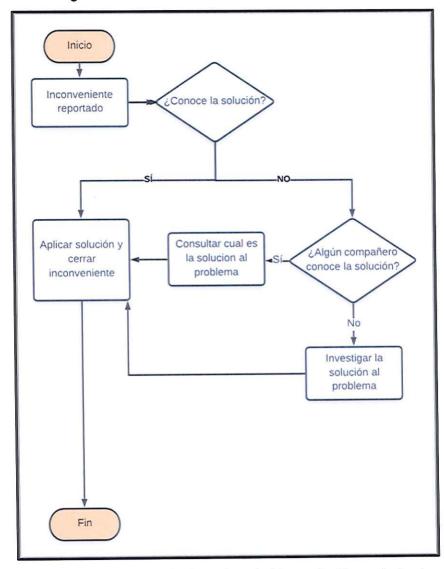


Figura 1. Flujo de trabajo antes de Base de Conocimiento.

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor
	Supervisor.	TAPTA:	Organizacional:
MA	Dr. SERGIO DANIEL CON	DE / JEG	Some to
	MON EN FSICOLO		14304



Práctica Profesional Supervisada (PPS)

Página 14 de 72

A fin de solventar la problemática encontrada, se solicita la realización de un sistema que logre recopilar las investigaciones realizadas por los analistas de la empresa con el objetivo de facilitar el acceso a las soluciones una vez que se hayan documentado.

Sobre la base de ese requerimiento, se coordinó la realización de una aplicación web que debe cumplir el objetivo presentado anteriormente, apegándose a varios requisitos planteados en los siguientes puntos del documento.

Metodología de trabajo

Como metodología de trabajo se siguió el ciclo de vida sugerido por el Modelo lineal secuencial, más conocido como Modelo en cascada. Este tipo de ciclo de vida propone un enfoque sucesivo para el desarrollo de software, iniciando desde un nivel de sistemas para luego continuar con análisis de los requisitos, diseño del software, desarrollo del código, pruebas del programa y el mantenimiento.

Este enfoque modelado con base en el ciclo de ingeniería convencional se encuentra comprendido por una serie de actividades que se detallan a continuación:

- Ingeniería y modelado de Sistemas/Información. Implica tener una visión analítica sobre el sistema general en el cual se emplaza el software a desarrollar, estimando los requisitos de cada módulo del sistema y asignando algunos de estos sub-requisitos del desarrollo, cómo pueden ser la forma de comunicación entre sus módulos, el hardware y la interacción con las personas o usuarios.
- Análisis de los requisitos del software. Se caracteriza por un análisis general en el que se detectan los requisitos de software con base a los requerimientos de los usuarios finales.

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Firma tutor
Organizacional:



Práctica Profesional Supervisada (PPS)

Página 15 de 72

- Diseño. Es un proceso extenso que se enfoca en varios atributos del programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental. Esta fase de diseño permite modelar la arquitectura del software exacta para cumplir con los requerimientos del cliente, permitiendo diferenciar cuales son los requisitos puntuales de cada módulo que integra este sistema.
- Generación de código. Fase en la que se inicia el desarrollo del código necesario para cumplir con los requisitos de software definidos en las etapas anteriores.
- Pruebas. Luego de generar el código, se prosigue con las pruebas de software. Este proceso se enfoca en testear los procesos lógicos internos además de las pruebas funcionales que validan la correcta ejecución de las interacciones con el usuario final.
- Mantenimiento. Luego del cierre de las fases de desarrollo y pruebas es posible que surjan modificaciones. Estos cambios, tanto sean por errores, fallas o mejoras se aplican posteriormente.

Firma Estudiante:

Firma Docente Supervisor:

TAPTA:

Firma docente tutor

Dr. SERGIO DANIEL CONDE

DOCTOR EN PSICOLOGIA SOCIAL



Práctica Profesional Supervisada (PPS)

Página 16 de 72

Requisitos del sistema

Antes de comenzar con el desarrollo se debe contar con una especificación en la que se indiquen las capacidades que el sistema necesita para cumplir con los objetivos planteados. Para esto se realiza lo que comúnmente se llama "Análisis de requisitos", en el cual se reconoce el problema planteado y se definen las necesidades de sistema que muestra la lista de condiciones necesarias para solventar la problemática indicada por la empresa.

La forma de trabajo a utilizar se asemeja con el método de Pressman² para la etapa actual de análisis. En esta metodología el proceso analítico se divide en cinco etapas:

- Reconocimiento del sistema: Etapa enfocada la definición de especificaciones del sistema, en la cual se debe contar con una buena comunicación entre los interesados del proyecto para detectar cada elemento del problema y cómo mejorar la experiencia del usuario.
- Evaluación y síntesis: Etapa basada en el flujo y estructura que debe tomar la información, estableciendo los diferentes factores que repercuten en el desarrollo como también así las restricciones que son impuestas.
- Modelización: Etapa asociada a la generación de modelos o mockups que funcionan como maqueta para entender el funcionamiento de los sistemas desarrollados, pudiéndose utilizar esta plantilla como base para agregar las funcionalidades finales.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor

TAPTA:

²Se conoce como "método de Pressman" a la metodología para el análisis de requisitos propuesta por el ingeniero de software Roger S. Pressman.



Práctica Profesional Supervisada (PPS)

Página 17 de 72

- Especificación: Etapa centrada en realización de representaciones del software para poder validar si se comprendieron las necesidades del sistema solicitado e identificar los casos de uso que deben ser agregados en la documentación.
- Revisión: Última etapa basada en una tarea iterativa en conjunto con los interesados del proyecto en la cual se debe comprobar que el sistema a desarrollar cumple con las necesidades y que serán satisfechas de forma eficiente.

Sobre la base en la aplicación de esta metodología se definieron los siguientes requisitos que debe cumplir nuestro sistema:

Requisitos Funcionales

- Creación, edición y modificación de artículos
- Exportación a PDF

Requisitos No Funcionales

- Interfaz mobile/responsiva³
- Fácil utilización
- Soporte de campos de texto enriquecido⁴

Firma Estudiante:

Firma Docente
Superviser:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:

³Se conoce como interfaz responsiva o *responsive* a las interfaces de usuario que se adaptan de forma dinámica a las diferencias distribuciones o tamaños de pantalla.

⁴ Texto enriquecido es aquel que a diferencia del texto plano, permite la inserción de varios detalles como tipografías, viñetas, cuadros, etc.



Práctica Profesional Supervisada (PPS)

Página 18 de 72

- Certificado SSL⁵
- Integración con el sistema de gestión RSA Archer

Arquitectura y tecnologías a utilizar

Previo al desarrollo de nuestro sistema, se debe hacer un análisis de los recursos y activos de la empresa para determinar la forma más eficiente de desplegar una aplicación como la solicitada.

En la primera parte se deben detectar los servidores disponibles en la empresa, para obtener un paneo general de la distribución de equipos y su disposición en la red corporativa.

La parte final contará con el análisis de los datos obtenidos para proponer el emplazamiento y la utilización de recursos ideales para el despliegue del sistema.

Arquitectura actual

El análisis de la arquitectura actual contempla el censado de los distintos dispositivos, redes y aplicaciones que se utilizan en la empresa, notando solo aquellos que son relevantes para el desarrollo de nuestro proyecto.

Actualmente, la arquitectura con la que se cuenta en la compañía está basada en máquinas virtuales⁶ (VM, Virtual Machine) desplegadas en varios

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor

TAPTA:

⁵ Se menciona un certificado SSL como un certificado que implementa una capa de securización de los datos realizando una encriptación cuando se envían hacia el servidor, de forma tal que si algún tercero los intercepta no sea capaz de entenderlos.

⁶ Se llama Máquina Virtual o Virtual Machine, VM a una representación virtual o emulada de un sistema operativo invitado se ejecuta dentro de un sistema operativo anfitrión o bien puede ser ejecutada de forma casi paralela compartiendo librerías del anfitrión hacia el invitado.



Práctica Profesional Supervisada (PPS)

Página 19 de 72

hipervisores⁷ VMware ESXi, mayormente utilizando el sistema operativo Windows Server 2012 R2. Estas VM cuentan con un enlace a la red principal, lo que permite situar en ese espacio los sistemas productivos de la empresa que son utilizados por los miembros de la misma.

Una de las restricciones encontradas en el análisis de requisitos es la integración con el sistema RSA Archer, utilizado para la gestión de la firma. Este sistema está implementado dentro de la arquitectura de máquinas virtuales mencionada en el párrafo anterior y comprende una arquitectura de un servidor de aplicaciones web IIS en conjunto con una base de datos MS SQL, detalles a tener en cuenta para la definición de la arquitectura en la que se debe implementar el nuevo sistema de Base de Conocimiento.

Además, RSA Archer cuenta con varias interfaces de programación de aplicaciones (API, application programming interface) con un conjunto de funcionalidades. Estas APIs permiten generar una integración de forma sencilla entre esta plataforma y el sistema a desarrollar, abstrayendo el manejo de usuarios, roles y la creación a bajo nivel (base de datos) de los registros, entre otras tareas facilitadas.

Propuesta de Arquitectura a implementar

La arquitectura descubierta permite conocer el mejor emplazamiento para nuestra aplicación. En adición, se demuestra que es posible realizar una integración utilizando las herramientas brindadas por el sistema de gestión RSA Archer.

Sobre la base del análisis, se detecta que el sistema a desarrollar debe ser emplazado dentro de la red empresarial. En adición, debido a la flexibilidad ofrecida

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:

⁷ Se conoce como Hipervisor al sistema, que permite la creación y ejecución (a nivel de hardware o sistema operativo) de máquinas virtuales.



Práctica Profesional Supervisada (PPS)

Página 20 de 72

por el sistema de gestión para el desarrollo de aplicaciones y a la ventaja de contar con una API sólida, se propone generar la aplicación "Base de Conocimiento" dentro de RSA Archer, para luego consumir esta información desde una aplicación web externa basada en el patrón de diseño MVC implementado con Express JS.

Conociendo el sistema que se implementa y los servidores disponibles se emplaza la aplicación web compartiendo el servidor del sistema de gestión debido a que este cuenta con una alta capacidad debido a su criticidad. La figura 2 muestra la forma en la que el sistema interactúa: el servidor de aplicación hostea tanto el sistema RSA Archer, que sirve en el puerto 443 de HTTPS y se comunica con la base de datos MS SQL a través del puerto 1433; como también el entorno web de la Base de Conocimiento, que sirve en el puerto 4433 y se comunica en *loopback* con la API del sistema de gestión del puerto 443 mencionado anteriormente.

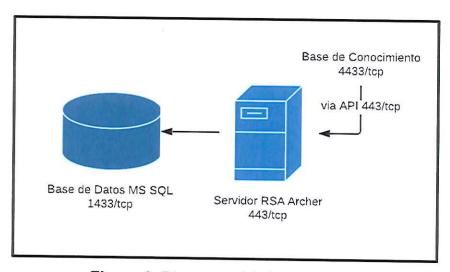


Figura 2. Diagrama del sistema a implementar.

Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL	Clay	Organizacional:
---	------	-----------------



Práctica Profesional Supervisada (PPS)

Página 21 de 72

Tecnologías a utilizar

A continuación, se listan las diferentes tecnologías a implementar para la realización de los distintos objetivos de este proyecto:

API (Application Programming Interface o Interfaz de programación de aplicaciones)

Las APIs componen un grupo de funciones, métodos, procedimientos y herramientas ofrecidas por una biblioteca que otorga una vía de interacción con un sistema dado. La ventaja que provee estas interfaces es la abstracción de las tareas internas que debe realizar el sistema para brindar el resultado esperado, evitando el conocimiento interno de cómo se implementan estas plataformas.

Para lo referido al proyecto, se utilizan dos tipos de APIs web, los Web Services SOAP y las APIs REST.

Web Services SOAP

Los Web Services SOAP son sistemas para compartir información a través de la red utilizando SOAP como protocolo para la transferencia de los mensajes.

Este tipo de API, todos los servicios disponibles y su forma de utilizarlos se definen mediante una interfaz Web Services Description Language, WSDL. Esta interfaz permite al servidor conocer la forma en la que se debe operar para contestar las peticiones por parte del cliente.

La información que se envía a través de estos Web Services se envían utilizando el protocolo SOAP, Simple Object Access Protocol, para

Firma Estudiante:

Firma Docente
Supervisor;

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Firma tutor
Organizacional:



Práctica Profesional Supervisada (PPS)

Página 22 de 72

garantizar extensibilidad en la seguridad y el routeo, neutralidad en el protocolo de transporte e independencia del modelo de información. Los datos enviados se formatean con XML, un lenguaje de marcado utilizado para generar una estructura de datos que puede ser entendida tanto por los usuarios como por las computadoras.

API REST

Las APIs REST implementan el estilo arquitectónico de software "Representational State Transfer" o Transferencia de Estado Representacional utilizado para permitir una interoperabilidad entre distintas aplicaciones desacopladas estableciendo una comunicación sobre el protocolo HTTP8.

El estilo REST no propone restricciones en su implementación a bajo nivel pero sí define seis principios arquitecturales que deben cumplirse en su interfaz final a alto nivel. Estas son:

1. Interfaz uniforme:

Plantea una definición uniforme de la forma en la cual se expondrán los recursos manteniendo las restricciones de identificación de recursos, manipulación de recursos a través de representaciones, mensajes autodescriptivos y *hypermedia* como motor del estado de la aplicación.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor TAPTA: /

⁸ Se llama protocolo de aplicaciones HTTP a aquel utilizado generalmente para la comunicación de sistemas web y es el protocolo fundamental para la comunicación de la World Wide Web.

⁹ Se llama Hypermedia a cualquier tipo de contenido que permita la interacción con los usuarios a través de la Web.



Práctica Profesional Supervisada (PPS)

Página 23 de 72

2. Cliente-Servidor:

Separa diferentes conceptos del sistema para proveer una interfaz simple y portable, independientemente de su implementación dentro del servidor.

Stateless¹⁰:

Promueve que las solicitudes hacia el servidor contengan absolutamente toda la información necesaria para que se pueda manejar las sesiones del lado del cliente.

4. Cacheable¹¹:

Requiere indicar si las respuestas de las solicitudes serán marcadas como cacheable o no cacheable para permitir la rápida reutilización de información en el cliente.

5. Sistema de capas:

Permite una arquitectura de diferentes niveles jerárquicos en los cuales la información sólo es accesible entre las capas inmediatas con las que se interactúa.

6. Codigo On Demand: (Opcional)

Amplía la funcionalidad del cliente permitiendo descargar y ejecutar código para reducir el número de funciones a implementar.

En REST, la forma de abstraer la información es a través de "recursos" identificados con su propio ID.

Firma Estudiante:

Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

DOCTOR EN PSICOLOGIA SOCIAL

 $^{^{10}}$ Se aplica el término Stateless a los sistemas que no se basan en estados o fases por lo que es posible que no se guarda información en las comunicaciones si no.

¹¹ Se denomina Cachear a la acción de realizar una "copia de acceso rápido" para cierta información que puede ser necesaria de forma recurrente.



Práctica Profesional Supervisada (PPS)

Página 24 de 72

La forma de representar estos recursos es a partir de información, metadata y enlaces hypermedia que ayudan al cliente a llegar al próximo estado deseado.

La solicitud de los recursos en el esquema REST se basa en la utilización de los métodos estándar que ofrece el protocolo HTTP:

GET

Recibir recurso

POST

Transmitir información

o PUT

Transmitir modificación parcial o total

información

PATCH

Transmitir modificación

total

de

información

DELETE

Eliminar información

JSON (JavaScript Object Notation)

JSON es una notación objetos utilizada como estándar para transportar información almacenada en objetos con una estructura de clave y valor. Esta notación es utilizada por lo general para la comunicación vía APIs REST y es comprendida por cualquier lenguaje de programación por lo que, a pesar de su nombre (Notación de Objetos de JavasCript), se considera independiente del lenguaje.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor

TAPTA:

Firma tutor

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 25 de 72

RSA Archer

La suite RSA Archer es un software de gestión corporativo¹² que se enfoca en la administración eficiente del *Governance* (Gobierno), Riesgo y Cumplimiento empresarial (GRC)¹³ implementando procesos que permiten el flujo continuo de la información dentro de las distintas áreas pertenecientes a la empresa, para una mejor utilización de esta.

Las ventajas de esta herramienta son:

- Flexibilidad a las adaptaciones: Permite realizar cambios en las aplicaciones desarrolladas sin contar con amplios conocimientos técnicos.
- Unificación de la información: Permite consultar todos los datos desde una fuente centralizada.
- Colaboración entre las áreas: Permite una coordinación entre las distintas áreas encargadas de trabajar con la información.

RSA Archer como plataforma cuenta con distintos componentes fundamentales para su funcionamiento:

- Aplicación web: Una web application que le otorga los usuarios una interfaz para trabajar con la herramienta.
- Base de datos de configuración: Una base de datos donde se guardan los parámetros de configuración, como también, información sobre el cliente y de sus aplicaciones.

Firma Estudiante:

Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:

¹² Se conoce como software de gestión empresarial al sistema que cuenta con varios módulos que permiten realizar el seguimiento y administración de las distintas áreas de una compañía.

¹³ La administración del Governance (gobierno), Riesgo y Cumplimiento (GRC) definen estos parámetros claves a gestionar en una empresa para cumplir con las tareas y alcanzar las metas u objetivos propuestos.



Práctica Profesional Supervisada (PPS)

Página 26 de 72

- Base de datos de instancia: Una base de datos que cuenta con todos los registros de información que se utilizan en una instancia o ambiente de la empresa.
- Repositorio de archivos: Un repositorio de ficheros que serán utilizados dentro de la aplicación.
- Servicios: Una serie de servicios que controlan el ambiente desplegado, administrando, por ejemplo, el motor de trabajos o los datos de configuración.

La aplicación web de esta suite tiene integrada una API lo que permite realizar desarrollos externos que manipulen la información alojada en las bases de datos de la plataforma para poder ofrecer, por ejemplo, otras interfaces a los clientes, o una automatización customizada en el manejo de los datos.

JavaScript

JavaScript es un lenguaje de programación interpretado implementado por lo general para la programación del lado del cliente o *clientside*¹⁴ dentro de los navegadores web, permitiendo agregar funcionalidad lógica a la interfaz visual, mejorando la usabilidad de estas.

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:

¹⁴ Se llama *clientside* a la ejecución de un programa en el lado del cliente, utilizado, por ejemplo, al ejecutar instrucciones dentro del navegador web de los usuarios finales.



Práctica Profesional Supervisada (PPS)

Página 27 de 72

Node.is

Node.js es un entorno de ejecución¹⁵ que permite la utilización de código JavaScript fuera del navegador brindando la posibilidad de programar el *backend* de aplicaciones web implementando este lenguaje de programación.

ExpressJS

Express es un marco de trabajo o *framework*¹⁶ para el desarrollo de aplicaciones web y *mobile* utilizando el entorno de ejecución de Node.js.

Este framework se caracteriza por contar con una lista básica de características web, permitiendo al desarrollador mantener una aplicación liviana con solo las funcionalidades necesarias.

Otra característica de ExpressJs es su flexibilidad implementa un flujo de *middlewares*¹⁷ y módulos de NodeJS para realizar diferentes funciones como: mapeadores objeto-relacional, segurización de las aplicaciones, exportación a pdf, etc. lo que permite tener una app Express modular, flexible y extensible.

Firma Estudiante:

Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

¹⁵ Se conoce como entorno de ejecución a un conjunto de herramientas que permiten la ejecución de un sistema dado, como por ejemplo la colección de un motor intérprete de JavaScript y bibliotecas para el despliegue de una aplicación web.

¹⁶ Se denomina *framework* a un conjunto de herramientas que proveen una estructura y módulos reutilizables (entre otras cosas) para el desarrollo de una solución tecnológica.

¹⁷ Se llama *Middleware* a una pieza de software encargada de realizar un procesamiento logico de información durante la comunicación hacia el sistema con el fin de adaptar una funcionalidad en medio de una transacción.



Práctica Profesional Supervisada (PPS)

Página 28 de 72

Como la mayoría de los *frameworks*, Express ofrece una estructura de directorios base para organizar nuestra aplicación.

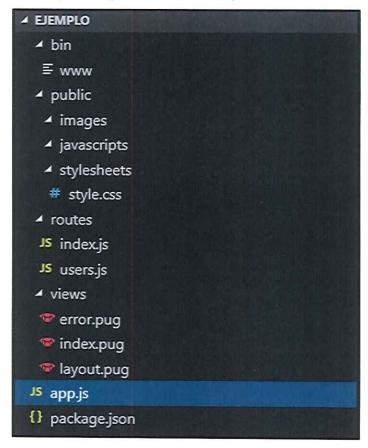


Figura 3. Esquema de un proyecto Express.js básico

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL
DOCTOR EN PSKOLOGIA SOCIAL



Práctica Profesional Supervisada (PPS)

Página 29 de 72

En la estructura de un proyecto ExpressJS se pueden notar los directorios relevantes para nuestro desarrollo:

Archivo app.js

El archivo app.js es el archivo principal de la aplicación, en la cual se indica que archivo de configuración utilizar, los permisos de la app, las rutas accesibles, etc.

Archivo package.json

El archivo package.json es el archivo que contiene todas los paquetes o dependencias de NPM que utiliza nuestro sistema.

Directorio bin

El directorio bin aloja el archivo "www", el cual se encarga de la configuración del servidor web de NodeJS y los parámetros extra que este necesita como los puertos a utilizar, los servicios a exponer, la segurización de la conexión, etc.

Directorio public

El directorio public es utilizado para alojar los archivos estáticos de la aplicación que estarán públicos para el acceso exterior. Estos archivos pueden ser tanto imágenes, así como plantillas de estilos CSS o archivos de JavaScript *clientside*.

Directorio routes

El directorio routes es utilizado para guardar los archivos "controladores" encargados de manipular las rutas de nuestra aplicación.

Estos controladores de rutas se representan con un archivo JavaScript que se utilizan para la resolución de solicitudes de los distintos

Firma Estudiante:

Firma Docente
Supervisor:
Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor TAPTA:



Práctica Profesional Supervisada (PPS)

Página 30 de 72

métodos de HTTP (GET, POST, DELETE, PUT, etc.) y debe ser asociado a una ruta definida dentro del archivo "app.js".

Directorio views

El directorio views es utilizado para alojar las diferentes vistas que se renderizan en la aplicación.

En el proyecto base se crean por defecto las vistas "layout.pug" utilizada como plantilla de estilos que se implementa en todas las vistas secundarias.

NPM

Npm es un sistema de gestión de paquetes¹⁸ para el entorno de ejecución Node.js. Esta herramienta permite la administración y de paquetes o bibliotecas que pueden utilizarse en nuestro proyecto.

A continuación, se listan los paquetes de Node.js requeridos para la Base de Conocimiento.

Express

Express es un *framework* de aplicaciones web de código abierto u *open-source* para Node.js que ofrece una estructura simple y flexible para el desarrollo tanto de aplicaciones web como móviles.

https://expressjs.com/en/resources/books-blogs.html

body-parser

body-parser es un middleware de Node.js utilizado para el parseo del contenido de los paquetes enviados por nuestra aplicación web,

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL
DOCTOR EN PSKOLOGIA SOCIAL

¹⁸ Se conoce como Paquete a un conjunto de programas y herramientas de software que se distribuyen de forma empaquetada y que contienen una cierta funcionalidad.



Práctica Profesional Supervisada (PPS)

Página 31 de 72

brindando una salida fácil y rápida a la hora de manejar e interpretar esta información.

https://github.com/expressjs/body-parser

compression

compression es un *middleware* de compresión para Node.js que es utilizado para reducir la carga que pueda tener el contenido de cada solicitud realizada aplicación. nuestra Este paquete forma parte de la suite de Express por lo que puede integrarse nativamente con las web applications que implementen este framework.

https://github.com/expressjs/compression

helmet

Helmet es una librería de Node.js que se encarga de manejar las solicitudes recibidas por nuestra aplicación con el fin de filtrar aquellas que están intencionadas a realizar algún tipo de ataque informático hacia nuestro sistema.

https://www.npmjs.com/package/helmet

Firma Estudiante:

Firma Docente Supervisor CONDE OCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor

TAPTA:

Firma tutor

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 32 de 72

o <u>morgan</u>

Morgan es una librería de Node.js cuya funcionalidad es servir como middleware para el *logging*¹⁹ de la información generada por nuestra aplicación, generando un trackeo de los eventos que suceden dentro de la misma.

https://github.com/expressjs/morgan

o <u>phantom</u>

Phantom es un paquete de Node.js que implementa la librería PhantomJS en el entorno de ejecución mencionado.

La funcionalidad de PhantomJS es permitir la manipulación del DOM²⁰ perteneciente a las vistas implementadas en nuestro sistema, brindando, por ejemplo, la captura de esta información en forma de imágenes o documentos pdf.

http://phantomjs.org/

o pug

Pug, antiguamente Jade, es un motor de *frontend*²¹ de alta performance que es implementado en Node.js.

https://www.npmjs.com/package/pug

Firma Estudiante:

Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL
DOCTOR EN PSKOLOGIA SOCIAL

¹⁹ Se conoce como *logging* o a la práctica de extraer información sobre el procedimiento llevado a cabo por un programa con el fin de poder hacer un seguimiento ante errores.

²⁰ Se llama DOM al Modelo de Objetos del Documento que representan la estructura de una interfaz de usuario, como puede ser la estructura de componentes de un sitio web HTML.

²¹ Se conoce como *Frontend* a la programación de interfaces para el usuario final.



Práctica Profesional Supervisada (PPS)

Página 33 de 72

o request

Request es un paquete básico de Node.js que permite la realización de solicitudes HTTP²² por parte de nuestra aplicación web hacia otros destinos.

https://github.com/request/request

sync-request

Request es un paquete básico de Node.js que permite la realización de solicitudes HTTP sincronas por parte de nuestra aplicación web hacia otros destinos.

https://github.com/ForbesLindesay/sync-request

jQuery

jQuery es una biblioteca de JavaScript que simplifica la manipulación de los elementos HTML, la modificación de los estilos, el manejo de interaccione con el usuario, la inclusión de animaciones y las solicitudes sin refrescar la página utilizando AJAX²³.

Esto se logra al utilizar una API fácil de implementar que puede trabajar a través de los diferentes navegadores web.

Firma Estudiante:

Supervisor:

Dr. SERGIO DANTEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL
DOCTOR EN PSKOLOGIA SOCIAL

²² Recibe el nombre de solicitud HTTP a la solicitud que se puede enviar hacia un servidor que entienda los distintos métodos del protocolo HTTP (GET, POST, DELETE, PUT, etc.).

²³ Se conoce como AJAX a la técnica de comunicación asincrónica que se ejecuta desde un cliente web hacia un servidor de forma que el cliente web pueda continuar con su trabajo mientras espera la respuesta a su solicitud.



Práctica Profesional Supervisada (PPS)

Página 34 de 72

<u>iQuery Soap</u>

Es un plugin que implementa la biblioteca jQuery para realizar solicitudes a los Web Services SOAP de forma sencilla. Este script permite enviar un SOAPEnvelope (envoltorio de una solicitud a Web Service SOAP) a través de AJAX, encapsulando la respuesta en forma de XML DOM, string XML o JSON.

MaterializeCSS

Es un framework de estilos CSS basado en estilo Material Design creado por diseñadores de Google. Esta biblioteca ofrece facilidades al momento de su implementación, con buena documentación y ejemplos de codigo, orientación hacia la mejora de experiencia de usuario UX y la aceleración del diseño web aplicando estilos predefinidos.

TinyMCE

Es una biblioteca *open-source* de JavaScript que permite la inserción de un editor de texto enriquecido a como campo de un formulario y expone una API para poder formatear y utilizar la información cargada en estos.

GIT

Git es un sistema de control versiones *open-source* para proyectos de software, lo que quiere decir que permite administración de los cambios que se realizan en el código desarrollado.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor TAPTA:



Práctica Profesional Supervisada (PPS)

Página 35 de 72

GitHub

GitHub es una compañia conocida por proveer servicios para alojar repositorios de Git en su sistema.

Desarrollo del sistema solicitado

En una primera instancia, teniendo de referencia la arquitectura sobre la que se desarrolla el sistema, se crea una Aplicación llamada "Base de conocimiento" dentro del sistema de gestión RSA Archer. Esta aplicación sirve de repositorio para los registros de artículos que se utilizarán en nuestro desarrollo y, conforme avanza el proyecto, será de utilidad para la realización de pruebas del sistema que son requeridas para la integración con nuestra WebApp.

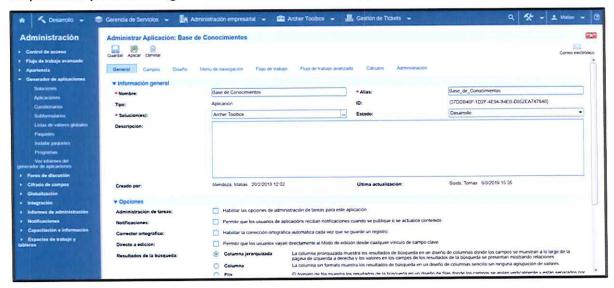


Figura 4. Configuración de aplicación "Base de Conocimiento" en RSA Archer.

Firma Docente	Firma docente tutor	Firma tutor
Dr. SERGIO DANIEL CONDE		Organizacional:
	Supervisor	,



Práctica Profesional Supervisada (PPS)

Página 36 de 72

Dentro de Archer, la aplicación de repositorio se generó con una serie de campos orientados a la eficiencia en la carga de datos y a la reutilización de valores ya creados en la plataforma. Estos campos se pueden observar en el diagrama de la Figura 5.

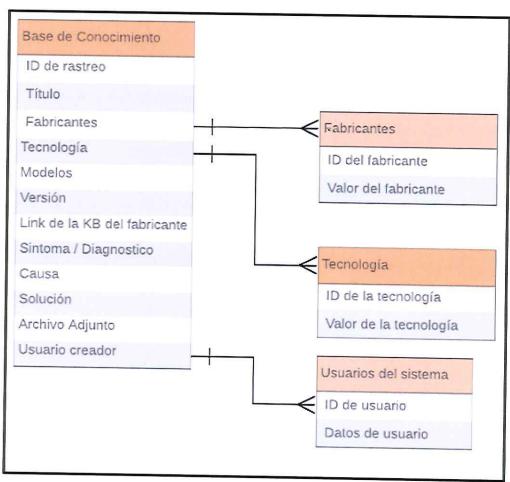


Figura 5. Diagrama de campos de "Base de Conocimiento" en RSA Archer.

Firma Estudiante:	Firma Docente Supervisor: Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL	Firma docente tutor TAPTA:	Firma tutor Organizacional:



Práctica Profesional Supervisada (PPS)

Página 37 de 72

El sistema de gestión implementado permite el desarrollo sencillo utilizando herramientas para crear campos indicando solo el nombre y tipo de dato que corresponde.

Además, se cuenta con un sistema de *drag and drop* que facilita la tarea de generar una interfaz para el usuario final. En la Figura 6 se observa el diseño o *layout* de la aplicación Base de Conocimiento dentro del sistema de gestión. Nótese que, además de los campos indicados en el diagrama de la Figura 5, se encuentran algunos extras como "Primera publicación" (fecha de creación), "Última actualización" y varios campos que no se muestran dentro de la plataforma debido a que se utilizan internamente para la comunicación con nuestra WebApp.

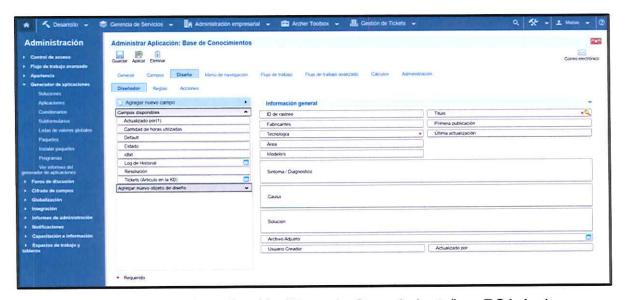


Figura 6. Campos de aplicación "Base de Conocimiento" en RSA Archer.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
Jada	Dr. SERGIO DANIEL CONDE DOCTOR EN PSECOLOGIA SOCIAL	Dav	forth



Práctica Profesional Supervisada (PPS)

Página 38 de 72

Con esta aplicación generada, solo resta hacer el sistema web con las llamadas a la API para que pueda ser accedido desde el exterior dando una capa extra de segurización a la WebApp.

Conociendo la estructura de este proyecto y la arquitectura deseada, se prosigue a la realización de pruebas a la API de RSA Archer.

Habiendo analizado la documentación del sistema de gestión, se detecta que la autenticación de las diferentes llamadas a la API se debe generar un "SessionToken" o token de sesión que se utiliza para identificar por un periodo de tiempo la sesión del usuario autenticado.

Para la generación de este token se debe realizar una solicitud HTTP POST utilizando en este caso el software Postman para este tipo de solicitudes, configurándolo tal como se describe en la documentación oficial del sistema de gestión:

Post:

/api/core/security/login

Headers:

"Accept":"application/json, text/html, application/xhtml+xml, application/xml; q=0.9, */*; q=0.8"

"Content-Type": "application/ison"

Body:

"InstanceName":{{InstanceName}},

"UserDomain":{{UserDomain}},

"Username":{{Username}},

"Password":{{Password}}

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor

TAPTA:



Práctica Profesional Supervisada (PPS)

Página 39 de 72

Como respuesta a la solicitud, Archer responde con el siguiente cuerpo:

```
Body:
 Links": [],
 "RequestedObject": {
    "SessionToken": "41C9A8F1FCB1865727C1124686885AF7",
    "InstanceName": "PROD",
    "UserId": 1205,
    "ContextType": 0,
    "UserConfig": {
      "TimeZoneId": "Argentina Standard Time",
      "TimeZoneIdSource": 1,
      "LocaleId": "es-AR",
      "LocaleIdSource": 2,
      "Languageld": 2,
      "DefaultHomeDashboardId": -1,
      "DefaultHomeWorkspaceId": -1,
      "LanguageIdSource": 1,
      "PlatformLanguageId": 2,
      "PlatformLanguagePath": "es-ES",
      "PlatformLanguageIdSource": 1
    },
"Translate": false,
    "IsAuthenticatedViaRestApi": true
 "IsSuccessful": true,
  "ValidationMessages": []"
```

Firma Estudiante:

Firma Docente Supervisor

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor TAPTA:

Práctica Profesional Supervisada (PPS)

Página 40 de 72

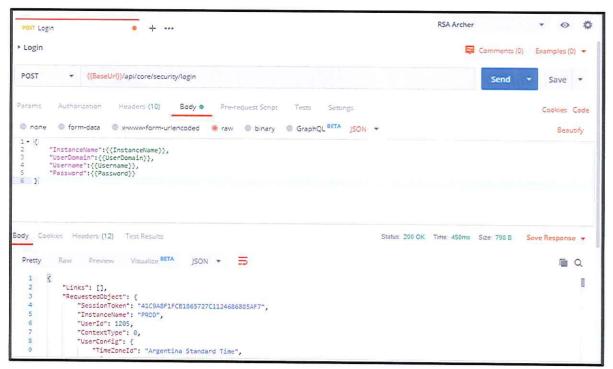


Figura 7. Cliente "Postman" realizando solicitud de loging.

Dentro de la respuesta se encuentra el objeto "RequestedObject" conteniendo el atributo "SessionToken" necesario para gestionar la autenticación del usuario al realizar diferentes consultas a la API.

Si bien se planificó la tarea de realizar un módulo en específico para la integración entre la aplicación web y RSA Archer, se detectó la posibilidad de realizar el desarrollo web en paralelo, agregando las consultas necesarias hacia la API del sistema de gestión, lo que permite validar las distintas partes de la Base de Conocimiento conforme se vayan desarrollando.

Firma Estudiante:	Firma Docente //	Firma docente tutor	Firma tutor
	Supervisor: //	TAPTA:	Organizacional:
Mh	Dr. SERGIO DANIEL CONDE	Duy.	Amb
		4/4	



Práctica Profesional Supervisada (PPS)

Página 41 de 72

El primer paso en el desarrollo web fue la utilización del utilitario "express-generator" para la creación de un proyecto base de Express que se utiliza como estructura para el proyecto. Esto se realiza ejecutando en línea de comandos la siguiente cadena: express --view=pug BaseDeConocimiento

Esto brinda una estructura base como la que se muestra en la Figura 3, con la cual organizar los distintos archivos que forman parte del sistema.

Teniendo en cuenta que la aplicación a desarrollar se basa en solicitudes HTTP es conveniente la generación de un helper que facilite la ejecución de las mismas. Para esto se desarrolla el helper "apiCalls.js" en el cual se encuentran varios parámetros que serán reutilizados dentro de la aplicación como por ejemplo la función "restApiCall" para las llamadas a la API, los *headers* para las solicitudes y los "cuerpos" para las llamadas de creación, edición y obtención de datos; entre otros atributos que ayudan a reducir la duplicación y *hardcodeo* de código.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN DISCOLOGIA SOCIAL Firma docente tutor TAPTA:

Práctica Profesional Supervisada (PPS)

Página 42 de 72

```
module.exports ={
     "Accept": "application/json,text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
     "Content-Type": "application/ison"
    "headersWAuth": function(sessionToken){
       "Accept": "application/json,text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
       "Content-Type": "application/json",
       "Authorization": "Archer session-id="+sessionToken
    3
        "headersSOAP": function(soapAction){
       "Content-Type" : "text/xml;charset=utf-8",
                   "SOAPAction" : soapaction
    }
  },
   "restAPICall" : function(headers, method, url, body){
      var response;
      options = {
          "headers": headers,
          "rejectUnauthorized": false
      if(!headers.SOAPAction){
          if (method==='POST' || method==='PUT'){
             options.body = JSON.stringify(body);
          } else {
              options['headers']['X-Http-Method-Override'] = 'GET';
      else(
          options.body = body;
```

Figura 8. Porción de "apiCalls.js".

Interfaz responsiva

El primer requerimiento que se puede cumplir es el de implementar una interfaz fi o adaptable, lo que significa que nuestra aplicación pueda ser compatible con diferentes tamaños de pantallas como por ejemplo un teléfono celular, una tableta o el monitor de una computadora.

Firma Estudiante:

Supervisor:

Firma docente tutor
TAPTA:

Organizacional:

Or. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL



Práctica Profesional Supervisada (PPS)

Página 43 de 72

Con base al conocimiento y experiencia se opta por la utilización la biblioteca de estilos CSS MaterializeCSS que provee una gran variedad de elementos adaptables que están definidos para la generación del tipo de interfaces requeridas.

Para lograr una implementación eficiente de esta biblioteca, se genera una vista "layout" que contiene los archivos de estilo necesarios para importar MaterializeCSS. Además, incluye los elementos básicos de nuestra aplicación como lo son la barra superior, el contenedor principal y la barra inferior.

Este archivo es reutilizado en todas las futuras vistas del sistema ya que cuenta con todos los estilos y archivos necesarios para la interfaz responsiva que se busca.

```
doctype html
html
html
head

title= title
link(rel='stylesheet', href='/stylesheets/materialize.min.css')
link(rel='stylesheet', href='/stylesheets/summernote-lite.css')
link(rel='stylesheet', href='/stylesheets/style.css')
link(rel='stylesheet', href='/stylesheets/style.css')
link(rel='stylesheet', href='https://fonts.googleapis.com/icon?family=Material+Icons')
body

cheader>
chave
chav
```

Figura 9. Porción de código de la vista "layout.pug" con los estilos de la app.

Firma Estudiante:

Firma Docente
Supervisor:

Firma docente tutor
TAPTA:

Organizacional:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL



Práctica Profesional Supervisada (PPS)

Página 44 de 72

Sesión

Luego de verificar la funcionalidad de la API con los test, se realiza un conector con las bibliotecas de programación para las utilización de las APIs, request y sync-request para replicar la obtención del SessionToken dentro de nuestra WebApp.

En un principio, se realiza una conexión con RSA Archer con objeto de generar un token de sesión para la herramienta. Para lograr esto, la primera parte del desarrollo dentro del framework ExpressJS se basa en agregar la ruta "/" o "/login" que tiene como objeto manejar la aplicación y redirigir las solicitudes al sistema de gestión.

Una ruta en la aplicación es un espacio de la aplicación por el cual se puede resolver una solicitud HTTP, esto es que se pueda responder a los distintos métodos que expone el protocolo mencionado (GET, POST, DELETE, PUT, etc.) aplicando cierta lógica del lado del servidor para lograr una funcionalidad dada.

Las nuevas rutas se introducen dentro del archivo "app.js", instanciando y asignando el controlador que gestiona de la misma, como se nota en las Figuras 8. y 9.

En este prototipo se propone generar el controlador de ruta "login.js" para poder resolver peticiones HTTP GET en la cual se debe mostrar el formulario de inicio de sesión; y POST en la cual se debe evaluar la información enviada para aceptar o rechazar el inicio de sesión.

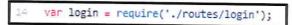


Figura 10. Instancia del controlador "login.js" dentro del archivo "app.js".

42 app.use('/login', login);

Figura 11. Asignación de la ruta "/login" al controlador "login.js".

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional;
Cons ju	Dr. SERGIO DANIEL CONDE	Collect,	fresh
			1//

Práctica Profesional Supervisada (PPS)

Página 45 de 72

El archivo controlador de ruta "login.js" cumple dos funciones:

- 1. Resolver solicitudes GET retornando la vista "login.pug" con el formulario de inicio de sesión.
- Resolver solicitudes POST reenviando la información del formulario al sistema de gestión para intentar generar un token de sesión.

```
var express = require('express'):
   var api = require('../apiCalls');
   var router = express.Router();
   /* GET home page. */
   router.get('/', function(req, res, next) {
     if (!req.query.st){
       res.render('login', { title: 'Login - Base de Conocimiento', logout: true});
        res.render('login', { title: 'Login - Base de Conocimiento' });
   });
13 router.post('/', function(req, res, next) {
       body = {
           "InstanceName":api.instanceName,
           "Username": req.body.username,
            "UserDomain":"",
            "Password": req.body.password
       response = api.restAPICall(api.headers, 'POST', api.url+'/api/core/security/login', body);
       res.send(response);
    77:
    module.exports = router;
```

Figura 12. Controlador de ruta "login.js".

Tal como se ve a partir de la línea 6 en la Figura 12, al realizar una solicitud GET, este controlador intentará renderizar la vista "login" con el formulario de sesión. Esta vista es corresponde al archivo "login.pug" del directorio "views" dentro de nuestro proyecto, la cual extiende a su vez la plantilla "layout.pug" para heredar los estilos definidos de forma global para la aplicación, como también la biblioteca de estilos

Supervisor: TAPTA:	
Supervisor.	Organizacional:
TANIEL CONDE	1/1
Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL	174



Práctica Profesional Supervisada (PPS)

Página 46 de 72

como MaterializeCSS que provee varias herramientas de diseño y permite solventar el requerimiento de una interfaz responsiva.

```
extends layout
block content
 <h3>Base de conocimiento</h3>
 <form action="javascript:void(0)" id="login">
 <div class="form-group">
 <label for="username">Usuario</label>
  <input type="text" class="form-control" id="username" placeholder="Ingrese su nombre de usuario" required>
  <div class="form-group">
  <label for="password">Contraseña</label>
  <input type="password" class="form-control" id="password" placeholder="Contraseña" required>
  <button type="submit" class="light-blue darken-4 waves-effect waves-light btn" id="submit">Loguear</button>
  </form>
block script
  script(src='/javascripts/login.js')
  if msg
      script(type='text/javascript').
        Materialize.toast('#{msg}', 3000, 'rounded');
```

Figura 13. Código de la vista "login.pug" extendiendo la plantilla "layout.pug".

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor TAPTA:



Práctica Profesional Supervisada (PPS)

Página 47 de 72

Por otro lado, se puede observar que el controlador realiza una instancia del helper "apiCalls" utilizado para la realización de solicitudes HTTP hacia RSA Archer, a su vez se puede notar a partir de la línea 13 de la Figura 12 cómo se realiza el cuerpo y la solicitud de logueo hacia el sistema de gestión tal como se realizó anteriormente en la prueba con el cliente Postman. La aplicación maneja esta solicitud del lado del cliente con un archivo JavaScript público "login.js" que se encuentra en el directorio "public/javascripts" y se encarga de realizar una consulta AJAX para generar el SessionToken.

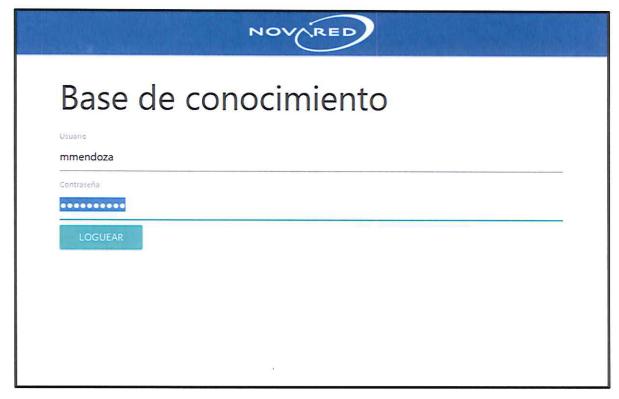


Figura 14. Vista "login.pug" mostrando el formulario de sesión.

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor
1	Supervisor:	TAPTA;	Organizacional:
(Mich	TANIEL CONDE	Thu	
	Dr. SERGIO D'ANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL		Month
			11/



Práctica Profesional Supervisada (PPS)

Página 48 de 72

Para el caso de una sesión iniciada de forma correcta, se agrega la ruta "/index" que solo verifica la existencia de un token de sesión, renderizando una vista simple de bienvenida en caso exitoso y redirigiendo a la ruta "/login" en caso de no contar con una sesión activa.

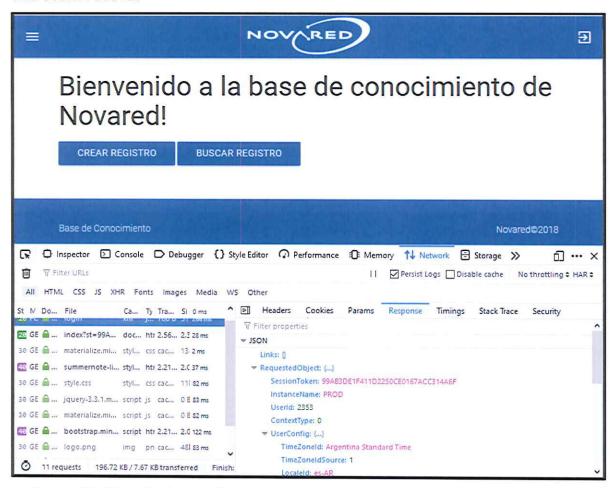


Figura 15. Vista "index.pug" mostrando la solicitud POST con el SessionToken.

Firma Estudiante:	Firma Docente Supervisor:	Firma docente tutor TAPTA:	Firma tutor Organizacional:
Class	Dr. SERGIO DANIEL CONE DOCTOR EN PSKOLOGIA SOCIAL	e CMY	Jan
			///



Práctica Profesional Supervisada (PPS)

Página 49 de 72

Alta de registros

Teniendo el conector que provee el SesionToken dentro del frontend como se nota en la Figura 15, se prosigue enviándolo como parámetro hacia una nueva ruta que gestione el Alta de registros dentro de la base de conocimiento.

Sobre la base del requerimiento Alta y visualización de un registro, se generan las rutas "/create" y "/register" manejadas por los controladores homónimos con las siguientes funcionalidades:

create.js

- Resolver solicitudes GET retornando la vista "create.pug" con el formulario de creación de registro.
- Resolver solicitudes POST reenviando la información del registro hacia el sistema de gestión y redirigir a una vista con el detalle del mismo en caso exitoso.
- 3. Redirigir a "/login" cualquier solicitud que no contenga un SessionToken activo.

register.is

- Resolver solicitudes GET con un parámetro "id" retornando la vista "register.pug" con la información del registro solicitado.
- 2. Redirigir a "/login" cualquier solicitud que no contenga un SessionToken activo.

Debido a la complejidad de los registros de Base de conocimiento, el controlador "create" debe realizar unas peticiones HTTP al sistema de gestión para poblar la lista de "Fabricantes" y "Tecnologías" y luego renderizar la vista "create.pug" que



Práctica Profesional Supervisada (PPS)

Página 50 de 72

cuenta con editores de texto enriquecido provistos por la biblioteca TinyMCE tal como lo estipulaba el requisito.

```
router.get('/', function(req, res, next) {

fabs = api.restAPICall(api.headersWAuth(req.query.st), 'GET', api.url+'/api/core/system/valueslistvalue/flat/valueslist/'+api.ids.fabVL

tecs = api.restAPICall(api.headersWAuth(req.query.st), 'GET', api.url+'/api/core/system/valueslistvalue/flat/valueslist/'+api.ids.tecVL

//areas = api.restAPICall(api.headersWAuth(req.query.st), 'GET', api.url+'/api/core/system/valueslistvalue/flat/valueslist/'+api.ids.ar

if(fabs.StatusCode === 401 || tecs.StatusCode === 401 /*|| areas.StatusCode === 401*/){

res.redirect('/logout?st='+req.query.st);

res.end();

}
```

Figura 16. Consulta por Fabricantes y Tecnologías en el controlador créate.



Figura 17. Editores de texto enriquecido TinyMCE.

La resolución de solicitudes POST por parte del controlador "create" es simple, solo se genera el body correspondiente para la solicitud de creación con la estructura

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor
	Supervisor:	TAPTA:	Organizacional:
(b)	TAMEL CONDE	() will	16
	Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL	(1)	ATTER
	DOCTOR ST.		



Práctica Profesional Supervisada (PPS)

Página 51 de 72

almacenada en "apiCalls.js" y se retransmite esto hacia el sistema de gestión. En caso de que el registro se cree de forma exitosa, se muestra un mensaje de tipo Toast (mensajes temporales) y luego se redirige al detalle del mismo con la ruta "/register?id=<Numero de Id>&st=<Token de Sesión>".

```
"getContentBody": function(reg, id = ''){
          body = {
               "Content":{
                   "LevelId" : this.ids.levelId,
55
                   "FieldContents" : {
                        [String( this.ids.tit)]: {
57
                            "Type" : 1,
                            "Value" : reg.titulo,
                              "FieldId": this.ids.tit
                         /*[String( this.ids.area)]: {
                              "Type" : 4,
                              "Value" : {
                                  "ValuesListIds" : [ reg.area],
                                  "otherText" : null
                                3,
                              "FieldId": this.ids.area
                          3,8/
                          [String( this.ids.fab)]: {
                               "Type" : 4,
                               "Value" : {
                                    "ValuesListIds" : [ reg.fab],
                                    "OtherText" : null
                                "FieldId": this.ids.fab
                            },
```

Figura 18. Sección del atributo getContentBody para la creación de un registro.

Firma Estudiante:

Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 52 de 72



Figura 19. Mensaje Toast post creación de registro.

El controlador "register" es el encargado de recibir la redirección post alta de un registro. Esta solicitud debe contener los parámetros "ld", correspondiente al identificador del registro en el sistema de gestión; y "st", correspondiente al token de sesión utilizado para las consultas hacia RSA Archer.

Como primer paso, dentro del controlador se intenta obtener todos los datos referidos al registro con una llamada GET a la API del sistema de gestión, en caso de obtenerlos exitosamente se procede a cargar los datos secundarios (Fabricante, Tecnología y Archivos Adjuntos) y luego enviar toda la información recolectada en un diccionario JSON para que pueda ser mostrado en la vista de detalle de registro "register".

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor
	Supervisor:	TAPTA:	Organizacional:
(M)	Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL	Duy	forth
		1	



Práctica Profesional Supervisada (PPS)

Página 53 de 72



Figura 20. Información de un registro post creación.

Firma Estudiante:

Supervisor:

PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Firma tutor
Organizacional:



Práctica Profesional Supervisada (PPS)

Página 54 de 72

Baja y modificación de registros

Lo siguiente es realzar las funcionalidades de baja y modificación. Para tal motivo se generan las rutas "/edit" y "/delete" que agregan la lógica y visual indicada para cumplir con estos requerimientos.

Para la adición de estas funciones se modifica la vista "register" agregando una sección con un formulario de edición que se oculta al inicio de la página y se muestra al presionar el botón de edición. Además, se agrega el botón que muestra un mensaje Toast preguntando por la eliminación y en caso de aceptar esa acción realiza la solicitud a la ruta /delete para la eliminación del registro. Los nuevos botones y los mensajes se pueden observar en la Figura 21.

Los controladores de rutas homónimos que se agregan cumplen las siguientes funciones:

edit.js

- Resolver solicitudes POST con el contenido del formulario de edición, ordenar la información con el formato de la API y reenviar esta al sistema de gestión.
- 2. Redirigir a "/login" cualquier solicitud que no contenga un SessionToken activo.

delete.is

 Resolver solicitudes GET con un parámetro "id" que se reenvía en una petición HTTP DELETE para la eliminación del registro identificado.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor

TAPTA:



Práctica Profesional Supervisada (PPS)

Página 55 de 72

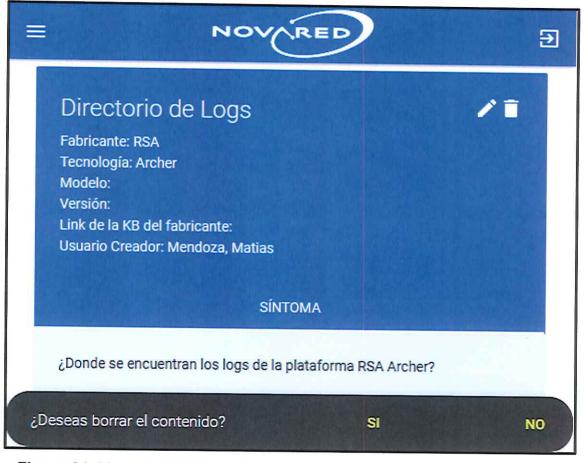


Figura 21. Mensaje Toast de eliminación y botones para Edición y Eliminación.

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Firma docente tutor
TAPTA:

Firma tutor
Organizacional:



Práctica Profesional Supervisada (PPS)

Página 56 de 72

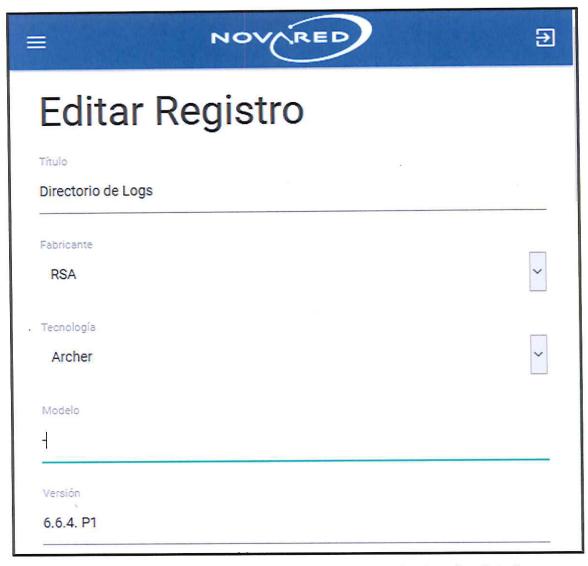


Figura 22. Formulario de Edición oculto en la vista "register".

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor
	Supervisor	TAPTA:	Organizacional:
m	Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL	July	Joseph Par



Práctica Profesional Supervisada (PPS)

Página 57 de 72

```
router.post('/', function(req, res, next) {
   body = api.getContentBody(req.body, req.body.id);
   options = {
     "url": api.url+'/api/core/content',
     "headers": api.headersWAuth(req.body.st),
     "method": 'PUT',
     "rejectUnauthorized": false,
     "body": JSON.stringify(body)
 var reque = request(options, function(error, response, body){
    if (!error && response.statusCode == 200) {
       var info = JSON.parse(body);
      if (info.IsSuccessful){
        res.render('register',{sessionToken: req.body.st, msg: 'Se ha editado el registro con exito', id: info.RequestedObject.Id});
       } else if (response.StatusCode === 401){
        res.redirect(401, '/logout?st='+req.body.st);
      ) else(
        res.render('search',{sessionToken: req.body.st, msg: 'Ha ocurrido un error al editar el registro.'});
    }else(
      console.log(error);
   });
```

Figura 23. Manejador de solicitudes POST en el controlador "edit".

```
router.get('/', function(req, res, next) {

content = api.restAPICall(api.headersMAuth(req.query.st), 'DELETE', api.url+'//api/core/content/'=req.query.id, '');

console.log(content);

if (content.Issuccessful){

res.render('create', { title: 'Crear - Base de Conocimiento', sessionToken: req.query.st, id: '', msg: 'Se ha eliminado el registro e
}

}

}

}
```

Figura 24. Manejador de solicitudes GET en el controlador "delete".

Búsqueda de registros

Luego de resolver el alta baja y modificación de los registros en nuestra WebApp, se prosigue a generar una ruta "/search" que debe mostrar un formulario con diferentes filtros para generar consultas de búsqueda dentro del total de registros que se manejan.

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 58 de 72

Previo a la implementación de este requerimiento, se detectó el método ofrecido por el sistema de gestión para realizar consultas es utilizar una llamada a sus WebServices SOAP a diferencia de todas las otras consultas ejecutadas a través de su API REST. Esta diferencia en la codificación llevada a cabo hasta el momento obligó a la búsqueda de alguna biblioteca que facilite el armado del body SOAP para la realización de estas llamadas.

La herramienta encontrada para tratar este cambio fue "jQuerySoap" una biblioteca de JavaScript para empaquetar solicitudes XML de SOAP dentro de solicitudes AJAX de jQuery. La complejidad de esta biblioteca es que solo funciona del lado frontend por lo que, para la resolución de este requisito, se planeó una vista de búsqueda que la incluya y que realice todas las consultas sin cambiar de ruta con solicitudes AJAX (como una Simple Page Application).

El controlador de ruta en este caso se encarga mostrar la vista de búsqueda "search" y reenviar las solicitudes POST hacia el sistema de gestión tal como se reciben (como un PROXY).

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor

TAPTA:

Firma tutor

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 59 de 72

```
']]>'];
                            console.log(xml.join(''));
             return xml.join('');
     function createRecordRow(record){
             html = ['',
                              '<a href="register?st='+localStorage.sessionToken='&id='+record.getAttribute('contentId')+'">'+record.child
                              ''+record.children[1].innerHTML+'',
                              ''+record.children[3].innerHTML+'',
                              ''+record.children[4].innerHTML+'',
                    '
            1;
            $('#tbody').append(html.join(''));
161 }
     function executeSearch (sessionToken, searchOptions, pageNumber){
            $.soap({
                    url: '/search',
                    method: 'ExecuteSearch',
                   SOAPAction: 'http://archer-tech.com/webservices/ExecuteSearch',
                   namespaceURL: 'http://archer-tech.com/webservices/',
                    appendMethodTouRL: false,
                           sessionToken: sessionToken,
                           searchOptions: searchOptions,
                           pageNumber: pageNumber
                    success: function (soapResponse) {
```

Figura 25. Fragmento del archivo público para la vista "search" que implementa jQuerySoap.

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor TAPTA:

Práctica Profesional Supervisada (PPS)

Página 60 de 72

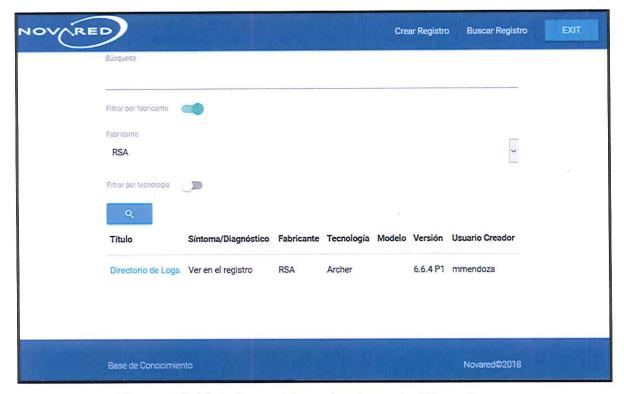


Figura 26. Vista "search" que implementa jQuerySoap.

Exportación a PDF de registros

Por último, se solicitó que el sistema web tenga la funcionalidad de exportar un registro de datos en formato PDF. Para tal requerimiento se agrega la ruta "/export", que muestra una vista limpia del detalle de un registro (sin información de sesión); y la ruta "/pdf", que ejecuta un método de la biblioteca "phantomJS" para capturar una pantalla web en la que se muestra el detalle de un registro y guarda esta captura en el formato solicitado.

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor
	Supervisor:	TAPTA:	Organizacional:
Jalon -	THE CONDE	Mary	1 6
	Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL	(Jacy	1294



Práctica Profesional Supervisada (PPS)

Página 61 de 72

Para la ejecución de esta funcionalidad se agrega un botón de "Exportación a PDF" en la vista register que detalla la información de un registro.

Figura 27. Manejador de solicitudes GET en el controlador "pdf" para exportar un registro a un archivo.

Firma Estudiante:

Firma Docente Supervisor: Firma docente tutor TAPTA:

Dr. SERGIO DANIEL CONDE

A:



Práctica Profesional Supervisada (PPS)

Página 62 de 72

Pruebas y ajustes del sistema solicitado

En esta sección se detallan las pruebas realizadas por el equipo para comprobar la funcionalidad básica de la aplicación tal como se indica en el plan de tareas a ejecutar para el proyecto:

- Realizar pruebas y ajustes sobre el módulo de integración con el sistema de gestión
- Realizar pruebas y ajustes sobre la aplicación web
- Realizar pruebas y ajustes integrales del sistema

La primera parte del proceso de pruebas se enfoca en el *testing* del módulo de integración con el sistema de gestión representado como el *helper* y las llamadas hacia las *API*s de RSA Archer.

Las pruebas funcionales de este conector se realizan con la ejecución de diferentes métodos y tipos de consultas hacia las interfaces que expone el sistema de gestión de GRC, comparando el resultado con el que se puede obtener al replicar las mismas llamadas desde el programa Postman.

La metodología de prueba que fue implementada en esta etapa consistió en:

- La creación de rutas temporales dentro de la aplicación, en las cuales se ejecutaron varias llamadas hacia la API REST y Web Services expuestos por RSA Archer mostrando en pantalla las respuestas obtenidas.
- La replicación de las mismas llamadas hacia las interfaces del sistema de gestión
- La comparación de las respuestas.

Firma Estudiante:

Supervisor:

Dr. SERGIO DANIEL CONDE

DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor TAPTA:

Firma docente tutor Organizacional:



Práctica Profesional Supervisada (PPS)

Página 63 de 72

Se definió como prueba exitosa el caso de que los resultados de ambas pruebas sean similares ya que se corroboraba el correcto funcionamiento del conector al obtener un resultado esperado para una consulta dada.

En la segunda instancia se realiza un conjunto de pruebas funcionales que engloban tanto la aplicación web y cómo el sistema completamente integrado.

La forma de trabajo aplicada para esta sesión de pruebas se basa en la ejecución repetida de varias tareas dentro de la aplicación, buscando que el comportamiento de esta sea el esperado para la función que se está buscando. Para tal fin se definieron una serie de casos de uso que engloban ciertas actividades que forman parte de la funcionalidad del sistema. Dichos casos de uso se pueden observar a continuación:

- Inicio de sesión
- Creación, edición y eliminación de un artículo
- Exportación de un artículo
- Búsqueda de artículos

En la planilla del Anexo 2 se detallan las pruebas funcionales realizadas por el equipo de desarrollo, indicando por caso de uso, una descripción de la prueba, la fecha de ejecución, el comportamiento esperado y el resultado o estado final de la prueba.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor

TAPTA:

Firma tutor

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 64 de 72

Conclusión

Luego de varias semanas de haber implementado el sistema y, acorde al feedback de los clientes y de los usuarios de la empresa, considero que el sistema cumple con los requerimientos iniciales ya que se logró resolver el inconveniente de una administración ineficiente de tiempos para al momento de encontrar los procedimientos a seguir para resolver los incidentes de la compañía.

Esto fue logrado gracias al desarrollo e implementación de una base de conocimientos centralizada que consolida las resoluciones encontradas por los técnicos.

Gracias a este sistema se notó una mejoría con respecto a la metodología anterior en la administración de los tiempos a la hora de resolver problemas. La ventaja obtenida en nuestra empresa se obtuvo dado a la facilidad que nuestra aplicación ofrece a la hora de buscar los artículos de su catálogo permitiendo filtrar todas las soluciones utilizando información clave, como por ejemplo: qué tecnología está relacionada, quién fue el fabricante o algún texto que identifique a los problemas. Por esto, tanto clientes como los usuarios internos están conformes con la funcionalidad implementada.

Posterior al despliegue del sistema se notaron algunos puntos que serán tenidos en cuenta como posibles mejoras para futuras *releases*. Dichos puntos se pueden observar a continuación:

 Manejo de sesiones nativas por parte del servidor, reemplazando el sistema actual que guarda el SessionToken del lado del cliente.

Firma Estudiante:

Firma Docente Supervisor: Firma docente tutor TAPTA:

Firma tutor
Organizacional:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Dilly



Práctica Profesional Supervisada (PPS)

Página 65 de 72

- Envío de notificaciones hacia usuarios claves de tecnologías para que estén al tanto de los nuevos artículos de la base de conocimiento que les interese.
- Implementación de un módulo para métricas y reportes para permitir un seguimiento tanto de la utilización como de las tendencias que surgen con base a nuevas tecnologías.

En la empresa se está trabajando en algunas de estas mejoras para completar un entregable con opción de realizar otras funcionalidades nuevas de forma evolutiva.

Reflexión sobre la Práctica Profesional Supervisada como espacio de formación

Después de haber finalizado la Práctica Profesional Supervisada puedo concluir que esta etapa propuesta por la universidad fue de gran provecho para mi ya que me permitió poder desarrollar dentro de un ambiente profesional todos los conocimientos que fueron incorporados durante la carrera.

Esta oportunidad es excelente ya que toda la experiencia que se gana al trabajar dentro de un ambiente profesional aporta muchos conocimientos para el desarrollo futuro de nuestra carrera.

Tanto el trabajo en equipo, la confianza en los líderes, la motivación para la autogestión de nosotros como recursos son herramientas claves que nos permiten progresar de forma rápida hacia el sin fin de oportunidades ofrecidas por el ámbito de la Ingeniería en Informática.

Considero que la Práctica Profesional Supervisada es clave para terminar de definir nuestro perfil porque nos permite trabajar sobre la rama de nuestra carrera con la que más afinidad tiene uno, en mi caso el desarrollo.

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSICOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Firma docente tutor
TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 66 de 72

Como puntos a destacar de esta asignatura me gustaría notar la forma de trabajo en equipo que, aunque fue practicada en materias de la cursada como Proyecto de Software, es una habilidad clave a desarrollar para fomentar nuestro crecimiento profesional.

Firma Estudianțe:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL

Firma docente tutor

TAPTA:

Firma tutor

Organizacionali



Práctica Profesional Supervisada (PPS)

Página 67 de 72

Bibliografía

- IEEE Standards Board, (1990), "IEEE Standard Glossary of Software Engineering Terminology"
 https://www.idi.ntnu.no/grupper/su/publ/ese/ieee-se-glossary-610.12-1990.pdf> (Consulta: 18/03/2019)
- Pressman, Roger, (2002), "Ingeniería del software, Un enfoque práctico"
- AA VV, (2019), "RSA Archer 6.6 Planning Guide for the Spanish language"
- AA VV, (15 de mayo de 2014), "¿Qué es API?"
 https://hipertextual.com/archivo/2014/05/que-es-api/ (Consulta: 18/03/2019)
- Maalej, Waleed; Robillard, Martin, (2012), "Patterns of Knowledge in API Reference Documentation",
 Shttps://mast.informatik.upi.hamburg.do/urp.content/upleade/2014/9

https://mast.informatik.uni-hamburg.de/wp-content/uploads/2013/03/TSE-2012-04-0081.R2 Maalej.pdf (Consulta: 18/03/2019)

- Yaapa, Hage, (2013), "Express Web Application Development"
- Chaffer, Jonathan; Swedber, Karl, (2013), "Learning jQuery Fourth Edition"
- Remy Blom, (2018), <https://github.com/doedje/jquery.soap> (Consulta: 18/03/2019)

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL Firma docente tutor TAPTA;



Práctica Profesional Supervisada (PPS)

Página 68 de 72

Anexos

Anexo 1. Otras alternativas posibles para el desarrollo de la Base de Conocimiento

En este espacio se mencionarán algunas alternativas posibles para el desarrollo del sistema requerido por la empresa, evitando algunas restricciones impuestas por la misma.

El sistema solicitado es lo que se conoce como una Base de Conocimiento, en la que se centralizan los artículos para solucionar algunos problemas recurrentes del trabajo habitual reduciendo, con este, los tiempos empleados en la resolución de incidentes solo con implementar los procedimientos que se encuentran en la base. Para el desarrollo del sistema mencionado se priorizaron algunos requerimientos propuestos por la empresa solicitante, principalmente la integración con el sistema de gestión utilizado, RSA Archer. Esta restricción conllevo a la necesidad de pensar en una solución a medida que este conectada al sistema de gestión, generando en la aplicación final una dependencia con todas las ventajas y desventajas del primero.

Al desestimar el requerimiento de la empresa, se podría desarrollar un propio sistema independiente basado en el patrón de diseño Modelo-vista-controlador (MVC) en el cual se cuente con el control total de la aplicación, evitando depender de las diferentes restricciones que introduce el sistema de gestión, administrando todos los aspectos que contempla una aplicación web como ser la conexión hacia la base de datos, la gestión de diferentes roles, el manejo de sesiones, los paneles administrativos, las altas, bajas o modificaciones de registros, etc.

A diferencia del sistema desarrollado, se deberá contar con toda una arquitectura nueva: un motor de base de datos acorde a la cantidad de transacciones requeridas,

Firma Estudiante:

Firma Docente
Supervisor:

Dr. SERGIO DANIEL CONDE
DOCTOR EN PSKOLOGIA SOCIAL

Firma docente tutor
TAPTA:

Organizacional:



Práctica Profesional Supervisada (PPS)

Página 70 de 72

De las opciones presentadas, como sistema de gestión de base de datos utilizaría PostgreSQL por la cualidad de ser un motor ampliamente probado en ambientes productivos como los de Uber, Netflix, Spotify o Instagram. Además de poseer una versión de código abierto que funciona de forma productiva.

De esta forma, la alternativa propuesta es una aplicación basada en el patrón de diseño implementado por el framework PHP, Laravel MVC que consumiría la información de varias tablas alojadas en una base de datos PostgreSQL y una interfaz implementada de forma nativa con las herramientas provistas por Laravel como puede ser basadas en su sistema de plantillas.

Firma Estudiante:

Firma Docente Supervisor

DI, SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL Firma docente tutor

TAPTA:



Práctica Profesional Supervisada (PPS)

Página 69 de 72

un servidor de aplicaciones dimensionado de forma correcta y un lenguaje de programación que garantice la solución de los requerimientos propuestos por el cliente.

El patrón propuesto podría implementarse utilizando un *framework* MVC que ya haya sido probado de forma productiva en el mercado tal como Spring MVC en el lenguaje Java, Laravel en el lenguaje PHP o Django en el lenguaje Python, entre otros *frameworks* que se utilizan actualmente de forma exitosa en el ámbito de tecnologías de la información.

Con base al conocimiento, optaría por desarrollar esta aplicación utilizando el framework Laravel debido a que nos provee muchas herramientas para facilitar las tareas de desarrollo como pueden ser el mapeador Objeto-Relacional (ORM, object-relational mapper) Eloquent, que permite hacer una abstracción de los registros de base de datos como objetos dentro del sistema; la consola de comandos Artisan, para la ejecución de algunas tareas recurrentes durante el desarrollo; la consola de comandos Tinker, para navegar a través de la base de datos utilizando sus registros como si fueran objetos; Laravel Nova, un generador de paneles de administración; y otros componentes.

Además de las herramientas mencionadas, Laravel tiene amplia utilización en el mercado, un gran ejemplo de esto es Pedidos Ya que lo utiliza dentro de su *stack* o pila de tecnologías para su conocida aplicación, además de páginas importantes como 9GAG o BBC que también utilizan este *framework* en sus sitios web.

Como motor de base de datos se pueden utilizar alternativas *open-source* como son MySQL o PostgreSQL o bien se podría optar por motores comerciales de la talla de Microsoft SQL Server u Oracle Database.

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL Firma docente tutor

TAPTA:



Práctica Profesional Supervisada (PPS)

Página 71 de 72

Anexo 2. Tabla de pruebas funcionales

Prueba	Descripción	Fecha de realización	Resultado esperado	Estado		
	1. Inicio de sesión					
1.1.	Iniciar sesión con credenciales válidas del sistema de gestión	10-oct.	Se genera un SessionToken en la URL y se redirige hacia la página de inicio.	Exitosa		
1.2	Iniciar sesión con credenciales inválidas del sistema de gestión	10-oct.	Se muestra un mensaje de error y el sistema no inicia sesión.	Exitosa		
	2. Creación, edició	n y eliminacio	ón de un artículo			
2.1.	Creación de artículo completando todos los campos	10-oct.	Se muestra un mensaje de creación exitosa y se redirige hacia una vista con el detalle del artículo.	Exitosa		
2.2.	Creación de artículo evitando campos requeridos	10-oct.	Se muestra un mensaje de error sobre los campos requeridos que no se encuentran completados.	Exitosa		
2.3.	Creación de artículo cargando un adjunto permitido	10-oct.	Se muestra un mensaje de creación exitosa y se redirige hacia una vista con el detalle del artículo con un botón para descarga de adjunto.	Exitosa		
2.4.	Creación de artículo cargando un adjunto no permitido	10-oct.	Se muestra un mensaje de error con forma de alerta en la cual se indica que el adjunto no está permitido.	Exitosa		
2.5.	Editar un artículo modificando todos los campos	10-oct.	Se realiza la modificación y se redirige hacia una vista de detalle del artículo con los nuevos cambios.	Exitosa		

Firma Estudiante:	Firma Docente	Firma docente tutor	Firma tutor,
	Supervisor:	TAPTĄ:	Organizacional:
/h	Dr. SERGIO DANIEL CONDE DOCTOR EN PSICOLOGIA SOCIAL	Open	Ja Ja



Práctica Profesional Supervisada (PPS)

Página 72 de 72

Prueba	Descripción	Fecha de realización	Resultado esperado	Estado		
2. Creación, edición y eliminación de un artículo (continuación)						
3.1.	Exportar un artículo en formato PDF	10-oct.	Se genera un enlace de descarga que permite obtener una archivo PDF con el detalle del registro.	Exitosa		
4. Búsqueda de artículos						
4.1.	Búsqueda global de artículos	10-oct.	Se listan todos los artículos de la base de conocimiento.	Exitosa		
4.1.	Búsqueda de artículos utilizando un filtro de fabricante	10-oct.	Se listan todos los artículos de la base de conocimiento que contengan el fabricante seleccionado.	Exitosa		
4.1.	Búsqueda de artículos utilizando un filtro de tecnología	10-oct.	Se listan todos los artículos de la base de conocimiento que contengan la tecnología seleccionada.	Exitosa		
4.1.	Búsqueda artículos utilizando una combinación de filtros	10-oct.	Se listan todos los artículos de la base de conocimiento que cumplan con los filtros introducidos.	Exitosa		

Firma Estudiante:

Firma Docente Supervisor:

Dr. SERGIO DANIEL CONDE DOCTOR EN PSKOLOGIA SOCIAL Firma docente tutor TAPTA:

