

Especificación de requisitos de software

Proyecto:

Revisión [Numero]

Instrucciones para el uso de este formato

Este formato es una plantilla tipo para documentos de requisitos del software.

Está basado y es conforme con el estándar IEEE Std 830-1998.

Las secciones que no se consideren aplicables al sistema descrito podrán de forma justificada indicarse como no aplicables (NA).

Notas:

Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.

Los textos entre corchetes del tipo “” permiten la inclusión directa de texto con el color y estilo adecuado a la sección, al pulsar sobre ellos con el puntero del ratón.

Los títulos y subtítulos de cada apartado están definidos como estilos de MS Word, de forma que su numeración consecutiva se genera automáticamente según se trate de estilos “Titulo1, Titulo2 y Titulo3”.

La sangría de los textos dentro de cada apartado se genera automáticamente al pulsar Intro al final de la línea de título. (Estilos Normal indentado1, Normal indentado 2 y Normal indentado 3).

El índice del documento es una tabla de contenido que MS Word actualiza tomando como criterio los títulos del documento.

Una vez terminada su redacción debe indicarse a Word que actualice todo su contenido para reflejar el contenido definitivo.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
[Fecha]	[Rev]	[Descripción]	[Firma o sello]

Documento validado por las partes en fecha: [Fecha]

Por el cliente

Por la empresa suministradora

Fdo. D./ Dña [Nombre]

Fdo. D./Dña [Nombre]

Contenido

Ficha del documento	3	
Contenido	4	
1	Introducción	6
1.1	Propósito	6
1.2	Alcance	6
1.3	Personal involucrado	6
1.4	Definiciones, acrónimos y abreviaturas	6
1.5	Referencias	6
1.6	Resumen	6
2	Descripción general	7
2.1	Perspectiva del producto	7
2.2	Funcionalidad del producto	7
2.3	Características de los usuarios	7
2.4	Restricciones	7
2.5	Suposiciones y dependencias	7
2.6	Evolución previsible del sistema	7
3	Requisitos específicos	7
3.1	Requisitos comunes de los interfaces	8
3.1.1	Interfaces de usuario	8
3.1.2	Interfaces de hardware	8
3.1.3	Interfaces de software	8
3.1.4	Interfaces de comunicación	8
3.2	Requisitos funcionales	8
3.2.1	Requisito funcional 1	9
3.2.2	Requisito funcional 2	9
3.2.3	Requisito funcional 3	9
3.2.4	Requisito funcional n	9
3.3	Requisitos no funcionales	9
3.3.1	Requisitos de rendimiento	9
3.3.2	Seguridad	9

3.3.3	Fiabilidad	9	
3.3.4	Disponibilidad	9	
3.3.5	Mantenibilidad	10	
3.3.6	Portabilidad	10	
3.4	Otros requisitos		10
4	Apéndices	10	

1

Introducción

El siguiente documento da a conocer la funcionalidad del sistema poli antifraude, el sistema externo que puede ser implementado en los sistemas existentes del poli (EPIC, moodle y CANVAS). Utilizando procesos básicos para identificar la identidad y prevenir el fraude académico en los parciales virtuales. *La introducción de la Especificación de requisitos de software (SRS) debe proporcionar una vista general de la SRS. Debe incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general del SRS.*

1.1 Propósito

Este documento de trabajo debería ser considerado como una descripción detallada del funcionamiento de la herramienta Poli-Antifraude, describe cada uno de los requisitos para su desarrollo, funcionamiento y aplicación. Así mismo puede ser utilizado como una guía para saber qué características, funcionalidades provee.

Esta guía está dirigida a cualquier persona que desee relacionarse más con la herramienta. Saber que variables, preguntas y problemas se consideraron en la decisión de requisitos, con el fin de tener una idea clara acerca de su uso.

§ Propósito del documento

§ Audiencia a la que va dirigido

1.2 Alcance

Poli-antifraude en este periodo de tiempo va a contener módulos de validación de identidad los cuales utilizarán sistema de preguntas de seguridad y reconocimiento por medio de fotos. Estos módulos se conectan directamente a la base de datos que se consolida al hacer evaluaciones virtuales. Se tomará en cuenta políticas de la universidad ya existentes como el pacto de honor para ofrecer la calidad en resultados de educación y una plataforma fiable para cualquier tipo de evaluación sugerida por la universidad.

§ Identificación del producto(s) a desarrollar mediante un nombre

§ Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir

1.3 Personal involucrado

Nombre: Julio Rodriguez

Rol: Product Owner*, Project Manager

Responsabilidades: Comunicación con las necesidades del cliente, Requerimientos funcionales y no funcionales, Definir casos de uso, Historia de usuario, Requerimientos de negocio

Información de contacto: 311-8546506 , basespoli@gmail.com

Nombre: Andres Buitrago

Rol: Scrum Master*, Project Manager

Responsabilidades: Control y asignación de tareas, Control y asignación de recursos, Control de cumplimiento, Optimización, PERT/CPM, Definir fases.

Información de contacto: 310-2199997, buitrago001@gmail.com

Nombre: Sebastian Canastero

Rol: Diseñador*, Tester

Responsabilidades: Diseño (UML, Mockups, Especificación de diseño funcional y técnico, Casos de pruebas), Crear ambiente de pruebas(Reportes de certificación, Pruebas unitarias y funcionales, Pruebas de despliegue)

Información de contacto: 310-3296631 , sebas91@gmail.com

Nombre: Alejandro Daza

Rol: Desarrollador*, DBA, Tester

Responsabilidades: Lógica de negocio y desarrollo, Diseño e implementación de bases de datos, Crear ambiente de Pruebas.

Información de contacto: 310-8011284 ,alejodt995@gmail.com

Relación de personas involucradas en el desarrollo del sistema, con información de contacto.

Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requisitos, validaciones de seguimiento, etc.

1.4 Definiciones, acrónimos y abreviaturas

Angular2: (comunalmente llamado "Angular" o "Angular.js"), es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

Spring: Es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

ORM: El mapeo objeto-relacional (más conocido por su nombre en inglés, Object-Relational mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.

MVC: El modelo–vista–controlador es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

JAVA: El lenguaje numero uno de programación en 2014, donde se estima que hay mas 8 billones de dispositivos habilitados con esta tecnología. Siendo fácil aplicar la metodología de la programación orientada a objetos, patrones, además de sus extensas librerías. Hacen que java sea el lenguaje escogido para manejar los requerimientos del proyecto.

TypeScript:TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases.

Procesamiento de imagen: Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

Node.js: Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Twitter Bootstrap: Es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

JSON: Acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos.

HTML5: (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

CSS3: Hojas de estilo en cascada (o CSS, siglas en inglés de Cascading Stylesheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado². Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML;

Maven: Es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant (y en menor medida a PEAR de PHP y CPAN de Perl), pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

MySQL: Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo.

Encriptación/Cifrado: En criptografía, el cifrado es un procedimiento que utiliza un algoritmo de cifrado con cierta clave (clave de cifrado) para transformar un mensaje, sin atender a su estructura lingüística o significado, de tal forma que sea incomprensible o, al menos, difícil de comprender a toda persona que no tenga la clave secreta (clave de descifrado) del algoritmo.

SHA: El SHA (Secure Hash Algorithm, Algoritmo de Hash Seguro) es una familia de funciones hash de cifrado publicadas por el Instituto Nacional de Estándares y Tecnología (NIST).

Moodle: Es una herramienta de tipo Ambiente Educativo Virtual, un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Suplantación: Es una forma de engaño por la cual una persona diferente que el estudiante asignado una asignación o un examen lo completa.

REST: La Transferencia de Estado Representacional (en inglés Representational State Transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

Persistencia: En programación, la persistencia es la acción de preservar la información de un objeto de forma permanente (guardado), pero a su vez también se refiere a poder recuperar la información del mismo (leerlo) para que pueda ser nuevamente utilizado.

JPA: Java Persistence API, más conocida por sus siglas JPA, es la API de persistencia desarrollada para la plataforma Java EE.

Indentación: En los lenguajes de programación de computadoras, la indentación es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores, teniendo en cuenta que los compiladores o intérpretes raramente consideran los espacios en blanco entre las sentencias de un programa.

IDE: Un entorno de desarrollo integrado o entorno de desarrollo interactivo, en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.

1.5 Referencias

Referencia	Titulo	Ruta	Fecha	Autor
------------	--------	------	-------	-------

Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.

1.6 Resumen

Se muestra a continuación cómo se va desarrollar la herramienta Poli-antifraude, se aclara que tipo de sistema a realizar, lo que ofrece como herramienta, los usuarios, factores de cambio y evolución. Para poder así identificar los requisitos esenciales para asegurar su funcionamiento y calidad.

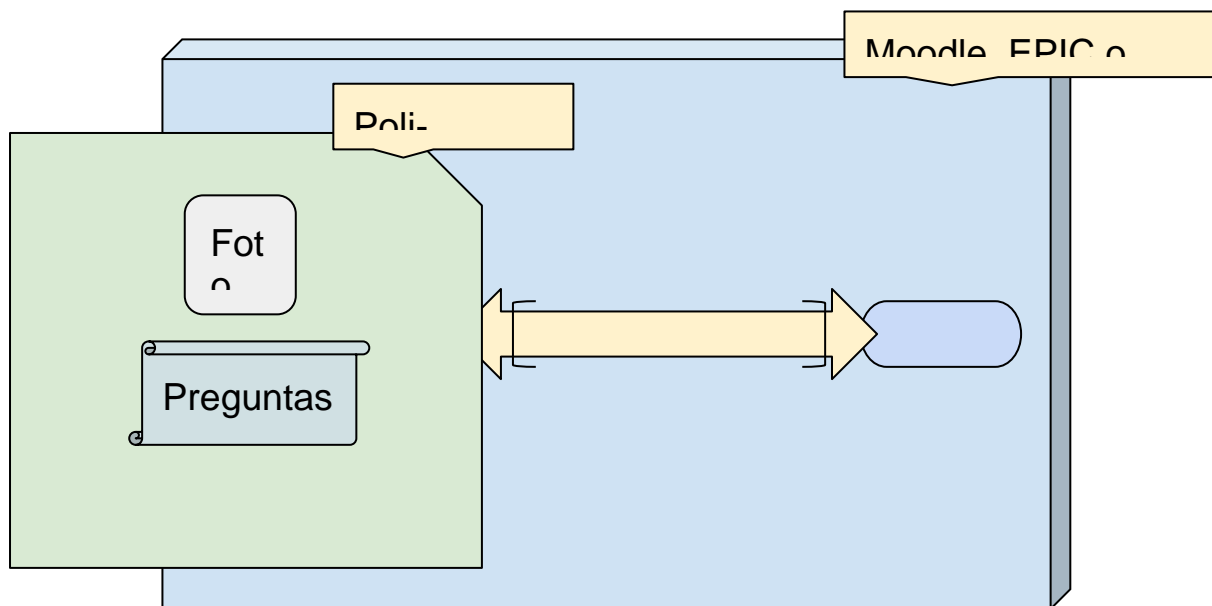
§ Descripción del contenido del resto del documento

§ Explicación de la organización del documento

2 Descripción general

2.1 Perspectiva del producto

La herramienta poli-antifraude va ser un sistema independiente que se conectará de forma embebida a los sistemas actuales de la universidad (Moodle, EPIC o Canvas). Hace la validación de el fraude académico y envía la información al examen que se esté presentando, para entender más lo que se desea hacer, puede ver el siguiente diagrama.



Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones facilita la comprensión.

2.2 Funcionalidad del producto

Poli-Antifraude tiene dos funcionalidades principales las cuales son hacer preguntas de seguridad estas son obtenidas de la base de datos del poli de información del estudiante, la cuales son sensibles y solo el estudiante las sabe, así aseguramos que otra persona no pueda responderlas, después de obtener estas preguntas, al inicio de la herramienta se realiza un cuestionario inicial para generar un código de patrón y al final del examen la herramienta también realiza el cuestionario y si hay un porcentaje específico para ese patrón se dará por válido el proceso de identificación.

La otra es tomar fotos y hacer comparación de ésta para identificar con el fin de verificar que no se hizo suplantación a la hora del examen, esta información es recopilada y enviada a moodle como una variable de fraude.

Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.

En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).

Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor pueda entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.

2.3 Características de los usuarios

Tipo de usuario: Estudiante

Formación: Cursando una carrera universitaria.

Habilidades: Ningún conocimiento sobre el manejo de la herramienta.

Actividades: El uso final de la herramienta, respondiendo a preguntas de seguridad y aceptando la captura de la foto.

Tipo de usuario: Profesor

Formación: Maestría , posgrado o especialización.

Habilidades: Entrenamiento básico de la herramienta.

Actividades: Profesor ingresa al sistema para verificar y ver la prueba de fraude. Además de configurar el sistema para comenzar en la misma fecha del examen.

Tipo de usuario: Administrativo

Formación: Tecnólogo, Pregrado o especialización.

Habilidades: Entrenamiento completo de la herramienta.

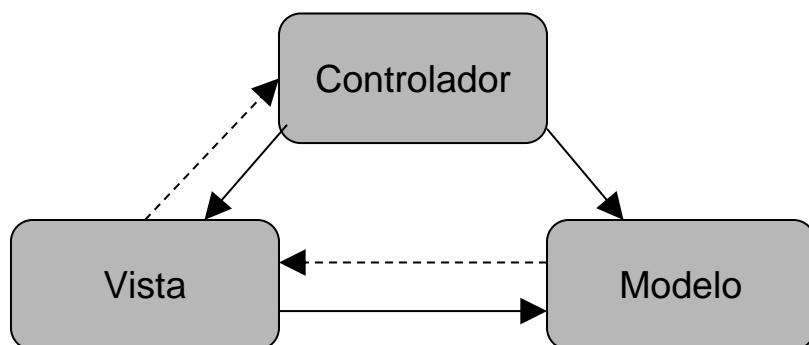
Actividades: Reportes generales de fraude (Cantidad de fraude por materia, cantidad fraudes por facultad, cantidad de fraudes global).

Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.

2.4 Restricciones

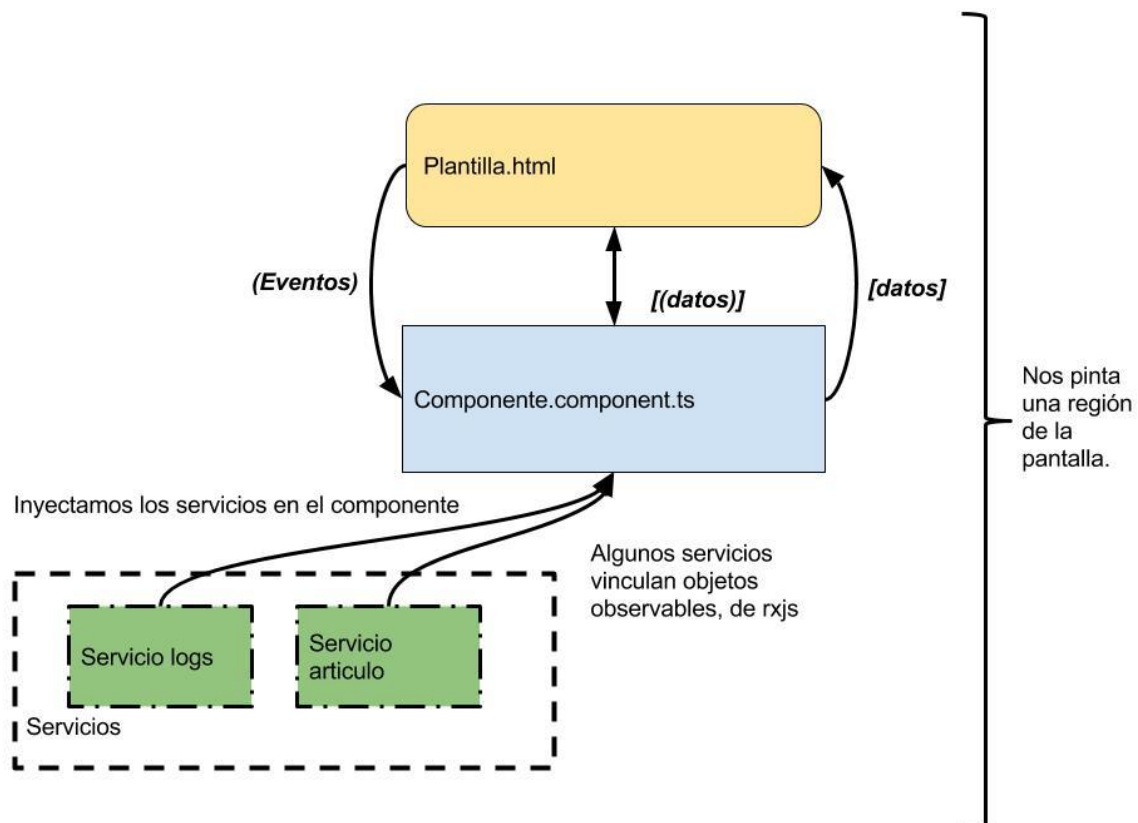
Restricciones de diseño.

Se determina el diseño va ser un patrón de arquitectura Modelo-Vista- Controlador.



Restricciones de implementación.

Se determina hacer el desarrollo utilizando una combinación de las tecnologías HTML5, CSS3 y JavaScript al utilizar los marcos de trabajo Twitter Bootstrap y TypeScript para templates e interfaces gráficas, Utilizando como frontend Angular 2 construido en un servidor node.js.



La lógica de negocio va a ser realizada utilizando tecnologías promovidas por la universidad como Java, utilizando como backend una aplicación REST en el marco de trabajo Spring que utiliza como persistencia JPA, construido en un servidor MAVEN.

Base de datos: MYSQL.

Restricciones de desarrollo.

Se determina utilizar la indentación estandarizada de cada lenguaje utilizado para su mayor entendimiento, además comentarios para su documentación.

Se utilizarán los siguientes IDE: sublime text, IntelliJ IDEA y Eclipse.

Se utilizará Karma para testear Angular2.

Restricciones de hardware.

Servidor

Procesador: 4 x 1.6 GHz CPU

RAM: 7 GB RAM

HDD: 1x40 GB para la base de datos y 1x40 GB para el software necesario para el despliegue.

Cliente.

Se debe tener en el hardware del cliente con al menos una cámara web de 5mp, cualquier explorador en su última actualización y conexión a internet constante.

Restricciones de uso

No se le permite al usuario final (estudiante) utilizar el click derecho para poder hacer alguna modificación en modo desarrollador del explorador.

Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.

2.5 Suposiciones y dependencias

Dependencia 1: Disponibilidad de la base de datos de los estudiantes del politécnico grancolombiano.

Dependencia 2: Disponibilidad de los servicios enlazados (Moodles, EPIC o Canvas).

Dependencia 3: Disponibilidad de los servidores de la herramienta.

Dependencia 4: El usuario debe tener una cámara web funcional.

Dependencia 5: Disponibilidad de la base de datos de la herramienta.

Dependencia 6: Conexión a internet.

Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo una asunción puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la SRS debería modificarse.

2.6 Evolución previsible del sistema

Gracias a que el alcance inicial no desborda los plazos de tiempo estipulados para la entrega de un producto de calidad, se consideraron mejoras para futuro como: Agregar código de verificación por celular, reconocimiento facial, patrones de comportamiento. También la construcción de un módulo completo que administre los exámenes para evadir cualquier falencia de seguridad que tenga moodle. Algo muy interesante que se está pensando a futuro es el poder utilizar el módulo desde su móvil y ofrece un servicio más portable con el slogan “educación a distancia y en cualquier lugar”.

Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro

3 Requisitos específicos

Aquí se describe, listan y detallan los requerimientos de sistema.

Esta es la sección más extensa y más importante del documento.

Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.

Los requisitos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (ej. RF 10, RF 10.1, RF 10.2,...).

Para cada requisito debe completarse la siguiente tabla:

Número de requisito		
Nombre de requisito		
Tipo	Requisito	Restricción
Fuente del requisito		
Prioridad del requisito	Alta/Esencial	Media/Deseado
Descripción:		

y realizar la descripción del requisito

La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.

3.1 Requisitos comunes de los interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

3.1.1 Interfaces de usuario

Número de requisito	RFIU1	
Nombre de requisito	Interfaz Sencilla	
Tipo	Requisito	
Descripción:	El sistema debe proveer una interfaz sencilla que no tenga muchos componentes cargados.	
Prioridad del requisito		Media/Deseado

Número de requisito	RFIU2
---------------------	-------

Nombre de requisito	Diseño Responsivo
Tipo	Requisito
Descripción:	La interfaz se debe adaptar al el tamaño de pantalla de que esté utilizando el usuario.
Prioridad del requisito	Baja/ Opcional

Número de requisito	<i>RFIU3</i>
Nombre de requisito	Marco de trabajo Bootstrap
Tipo	Requisito
Descripción:	La interfaz debe utilizar los componentes y estilos del marco de trabajo bootstrap.
Prioridad del requisito	Alta/Esencial

Número de requisito	<i>RFIU4</i>
Nombre de requisito	Canvas de procesamiento de imagen
Tipo	Requisito
Descripción:	El debe mostrar un canvas el cual se va usar para la captura de fotos y el procesamiento de esta.
Prioridad del requisito	Alta/Esencial

Número de requisito	<i>RFIU5</i>
Nombre de requisito	Interfaz independiente
Tipo	Requisito
Descripción:	La interfaz debe utilizar los componentes y estilos del marco de trabajo bootstrap.
Prioridad del requisito	Alta/Esencial

Describir los requisitos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz. Por ejemplo posiblemente el cliente ha especificado el estilo y los colores del producto. Describa exacto cómo el producto aparecerá a su usuario previsto.

3.1.2 Interfaces de hardware

Número de requisito	<i>RFIH1</i>
Nombre de requisito	Camara Web
Tipo	Requisito
Descripción:	El usuario final (estudiante) debe tener una camara web minimo de 5 mp.
Prioridad del requisito	Alta/Esencial

Número de requisito	<i>RFIH2</i>
Nombre de requisito	Periféricos
Tipo	Requisito
Descripción:	El usuario debe tener los periféricos generales (mouse y teclado).
Prioridad del requisito	Alta/Esencial

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

3.1.3 Interfaces de software

Número de requisito	<i>RFIS1</i>
Nombre de requisito	Integración del Software
Tipo	Requisito
Descripción:	El sistema debe integrarse con los otros sistemas vigentes como son el moodle y EPIC
Prioridad del requisito	Alta/Esencial

Número de requisito	<i>RFIS2</i>
Nombre de requisito	Explorador Actualizado
Tipo	Requisito
Descripción:	El usuario debe tener un explorador actualizado para poder utilizar la herramienta sin ningún problema

Prioridad del requisito

Media/Deseado

Número de requisito

RFIS3

Nombre de requisito

Disponibilidad de servidores

Tipo

Requisito

Descripción:

el servidor para el front-end node.js y el servidor para el back-end maven debe estar disponible cuando se haga uso de la herramienta.

Prioridad del requisito

Alta/Esencial

Proposito

Con el fin de poder construir el ambiente de desarrollo para el front-end y back-end respectivamente. De igual manera poder hacer el entorno de pruebas, y el entorno de despliegue.

Indicar si hay que integrar el producto con otros productos de software.

Para cada producto de software debe especificarse lo siguiente:

§ Descripción del producto software utilizado

§ Propósito del interfaz

§ Definición del interfaz: contiendo y formato

3.1.4 Interfaces de comunicación

Número de requisito

RFIC1

Nombre de requisito

Base de datos

Tipo

Requisito

Descripción:

Se debe tener una conexión a las bases de datos como de la herramienta y de la universidad para hacer proceso almacenados de CRUD y de lógica de negocio.

Prioridad del requisito

Alta/Esencial

Protocolo:

TCP/IP

Número de requisito

RFIC2

Nombre de requisito

Conexión cliente - servidor

Tipo	Requisito
Descripción:	Se debe tener una conexión desde el cliente al servidor conectado el front-end con su back-end.
Prioridad del requisito	Alta/Esencial
Protocolo:	HTTP requests (GET, POST)

Número de requisito	RFIC3
Nombre de requisito	Servicio - Servicio
Tipo	Requisito
Descripción:	Conexión de las lógicas de negocio de nuestra herramienta y el moodle para hacer actualización de base de datos.
Prioridad del requisito	
Protocolo:	HTTP requests (GET, POST), ORM

Describir los requisitos del interfaces de comunicación si hay comunicaciones con otros sistemas y cuales son las protocolos de comunicación.

3.2 Requisitos funcionales

Número de requisito	RF1
Nombre de requisito	Obtener la información del moodle
Tipo	Requisito
Fuente del requisito	Moodle
Prioridad del requisito	Alta/Esencial
Descripción:	Se obtiene la información de la base de datos que utiliza la base de datos de el poli para poder proveer con parámetros de entrada a la herramienta.

Número de requisito	RF1.1
Nombre de requisito	Obtener la información del estudiante.

Tipo	Requisito
Fuente del requisito	Base de datos universidad.
Prioridad del requisito	Alta/Esencial
Descripción:	Con la información del moodle se genera un proceso para obtener la información del estudiante para general las preguntas de seguridad.

Número de requisito	RF1.2
Nombre de requisito	Obtener información del examen.
Tipo	Requisito
Fuente del requisito	Base de datos moodle.
Prioridad del requisito	Alta/Esencial
Descripción:	Se obtiene la información del moodle para poder enlazar los procesos que genera la herramienta Poli-antifraude, recibiendo el código de la materia, el código del examen, fecha del examen, tiempo del examen, e intentos.

Número de requisito	RF2
Nombre de requisito	Generar el comunicado de aceptación de términos y condiciones.
Tipo	Requisito
Fuente del requisito	Poli-Antifraude
Prioridad del requisito	Media/Deseado
Descripción:	Se genera un comunicado para informar al estudiante de los procesos de validación de identidad que va hacer la herramienta, que reglas de la universidad se están cumpliendo y deberes del estudiante debe acordar, Recibe como entrada la aceptación por mediante de un botón.

Número de requisito	RF3
---------------------	-----

Nombre de requisito	Generar preguntas de seguridad
Tipo	Requisito
Fuente del requisito	RF1.1
Prioridad del requisito	Alta/Esencial
Descripción:	Con información suministrada por la base de datos de la universidad se genera 3 preguntas aleatorias de seguridad para verificar la identidad del estudiante se realiza al comienzo y al final del examen. REcibe como entrada la respuesta a cada preguntas mediante listas o checkboxes.

Número de requisito	RF4
Nombre de requisito	Mostrar Canvas
Tipo	Requisito
Fuente del requisito	<i>RFIH1</i> , Poli-Antifraude . RF3
Prioridad del requisito	Alta/Esencial
Descripción:	Se dibuja el canvas para poder hacer la captura de imagen y procesamiento de imagen. Entrada una booleano de la validación de preguntas de seguridad.

Número de requisito	RF4.1
Nombre de requisito	Tomar Fotos
Tipo	Requisito
Fuente del requisito	<i>RF4</i> , Poli-Antifraude
Prioridad del requisito	Alta/Esencial
Descripción:	Se toman fotos al inicio, en la mitad de el examen, un aleatoriamente en el examen y al final, se almacenan temporalmente en el canvas para realizar procesamiento de imagen y verificar la identidad del estudiante. Entrada fotos en formato png, y sale una variable de verificación (booleana).

Número de requisito	RF4.2
Nombre de requisito	Procesamiento de imagen.
Tipo	Requisito
Fuente del requisito	RF4.1, Poli-antifraude
Prioridad del requisito	Alta/Esencial
Descripción:	Se hace el procesamiento de las imágenes identificando propiedades esenciales de la imagen para poder así compararlas con el resto de ellas.

Número de requisito	RF3.1
Nombre de requisito	Procesamiento de preguntas.
Tipo	Requisito
Fuente del requisito	RF3, Poli-antifraude
Prioridad del requisito	Alta/Esencial
Descripción:	Se hace la comparación de las respuestas del estudiante y se considera un porcentaje de aprobación para que se valide su identidad y se considere como legítima la prueba realizada. Variable de confirmación.

Número de requisito	RF5
Nombre de requisito	Enviar Información
Tipo	Requisito
Fuente del requisito	RF1
Prioridad del requisito	Alta/Esencial
Descripción:	Se valida toda la información realizada por la herramienta y se consolida en la base de datos de la universidad, así como en la base de datos del poli.

Número de requisito	RF6
---------------------	-----

Nombre de requisito	Generar Reportes
Tipo	Requisito
Fuente del requisito	Poli anti-fraude
Prioridad del requisito	Media/deseado
Descripción:	Con la información de recopilada se generar reporte de fraude dando la opción al administrador filtrar los resultados obtenidos por materia, facultad, estudiante, corte, y demás.

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

En ellas se incluye:

§ Comprobación de validez de las entradas

§ Secuencia exacta de operaciones

§ Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)

§ Parámetros

§ Generación de salidas

§ Relaciones entre entradas y salidas (secuencias de entradas y salidas, formulas para la conversión de información)

§ Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

Las requisitos funcionales pueden ser divididos en sub-secciones.

3.2.1 Requisito funcional 1

3.2.2 Requisito funcional 2

3.2.3 Requisito funcional 3

3.2.4 Requisito funcional n

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

Todos estos requisitos deben ser medibles. Por ejemplo, indicando “el 95% de las transacciones deben realizarse en menos de 1 segundo”, en lugar de “los operadores no deben esperar a que se complete la transacción”.

3.3.2 Seguridad

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

- § Empleo de técnicas criptográficas.*
- § Registro de ficheros con “logs” de actividad.*
- § Asignación de determinadas funcionalidades a determinados módulos.*
- § Restricciones de comunicación entre determinados módulos.*
- § Comprobaciones de integridad de información crítica.*

3.3.3 Fiabilidad

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisible.

3.3.4 Disponibilidad

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

3.3.5 Mantenibilidad

Identificación del tipo de mantenimiento necesario del sistema.

Especificación de quien debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador.

Especificación de cuando debe realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanales y mensuales.

3.3.6 Portabilidad

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

- § Porcentaje de componentes dependientes del servidor.*
- § Porcentaje de código dependiente del servidor.*
- § Uso de un determinado lenguaje por su portabilidad.*
- § Uso de un determinado compilador o plataforma de desarrollo.*
- § Uso de un determinado sistema operativo.*

3.4 Otros requisitos

Cualquier otro requisito que no encaje en ninguna de las secciones anteriores.

Por ejemplo:

Requisitos culturales y políticos

Requisitos Legales

4 Apéndices

Pueden contener todo tipo de información relevante para la SRS pero que, propiamente, no forme parte de la SRS.