

UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA
FACULTAD DE INGENIERÍA DE MINAS, GEOLOGÍA Y CIVIL
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**“APLICACIÓN WEB PARA CONTROL DE ASISTENCIA DE ESTUDIANTES
EN EL CENTRO PREUNIVERSITARIO DE LA UNSCH, REGIÓN AYACUCHO,
2023”**

Tesis para optar el título profesional de:

INGENIERO DE SISTEMAS

PRESENTADO POR:

Bach. Elmer Tueros Puchuri

ASESOR:

Dr. Ing. Efraín Elías Porras Flores

Ayacucho - Perú

2023

Dedicatoria

Quiero comenzar dedicando todo mi esfuerzo a Dios que es parte de mi vida y siempre me ha guiado con la ayuda de mi familia siendo mis padres uno de las primeras personas que admiro cada día, mi padre Mauro Tueros López que está presente siempre entre mis decisiones apoyándome indistintamente y a mi madre Ricardina Lucila Puchuri Falconi que me ayudo a seguir mis sueños demostrándome que el nivel de lucha siempre se mejora, y a mis hermanos Ronald y Rooney Mauro que lograron ser buenos profesionales logrando ser una inspiración para mí en este proceso para terminar esta hermosa carrera de ingeniería de sistemas.

Agradecimiento

A mis padres y hermanos, porque han sido las personas más influyentes para ayudarme a alcanzar mis metas y hacer realidad mis sueños.

A mi enamorada Lizbeth Milagros Palomino Vega por poner su fe en mí y ser mi apoyo en las buenas y en las malas.

A la Universidad San Cristóbal de Huamanga por la fantástica oportunidad de avanzar en mi carrera

A los maestros y asesores que sentaron las bases para llevar mi aprendizaje y la aplicación de lo aprendido al siguiente nivel.

Al Dr. Ing. Efraín Elías Porrás Flores por aconsejarme y permitirme aprovechar su experiencia y conocimiento.

Resumen

El centro Preuniversitario de la UNSCH es una institución el cual tiene una trayectoria de más de 32 años en educación el cual es parte de la Universidad de San Cristóbal de Huamanga; gracias a esto, actualmente garantiza hasta un 70% de ingreso en universidades a lo largo del país. Es así como esta academia se ha convertido en una de las más reconocidas, contando con ventajas como: pruebas modelo para los exámenes de ingreso directo, 40% de vacantes directas a la Universidad de San Cristóbal para los estudiantes que posean el mejor desempeño, Una plantilla docente especializada con más de 20 años de experiencia en este modelo.

El proceso manual que usan no favorece un óptimo uso del tiempo en los procesos relacionados con el manejo de información de los estudiantes que asisten a las clases presenciales como a los exámenes del centro preuniversitario, por la cantidad inexacta de asistentes no se garantiza que la información que manejan sea exacta, valida y oportuna; para resolver estos problemas se implementará una aplicación web que lograra la gestión de asistencia de los estudiantes además dar solución a las principales actividades de la asistencia a clases como también al examen del centro preuniversitario, consiguiendo de esta manera una atención rápida al estudiante y manteniendo un estrecho control de asistencia.

Finalmente se cumple con el objetivo general como los objetivos específicos que consiste en automatizar actividades primordiales del control de asistencia; se ha logrado aplicar satisfactoriamente la metodología ágil de programación extrema logrando el desarrollo de las historias de usuarios, seguidamente se realizó con éxito las tareas de ingeniería esto permitió al usuario realizar las pruebas de aceptación que permite proceder con el desarrollo y ejecución de la aplicación del centro preuniversitario de la Universidad Nacional de San Cristóbal de Huamanga.

Palabras claves: Programación extrema, centro pre universitario, control de asistencia

Abstract

The UNSCH Pre-University Center is an institution which has a history of more than 32 years in education which is part of the University of San Cristóbal de Huamanga; Thanks to this, it currently guarantees up to 70% admission to universities throughout the country. This is how this academy has become one of the most recognized, with advantages such as: model tests for direct entrance exams, 40% of direct vacancies at the University of San Cristóbal for students with the best performance, a specialized teaching staff with more than 20 years of experience in this model.

The manual process they use does not favor an optimal use of time in the processes related to the management of information of the students who attend face-to-face classes such as the exams of the pre-university center, due to the inaccurate number of attendees it is not guaranteed that the information that they handle is accurate, valid and timely; To solve these problems, a web application will be implemented that will manage the attendance of the students, as well as provide a solution to the main activities of class attendance as well as the exam of the pre-university center, thus achieving rapid attention to the student and maintaining a tight attendance control.

Finally, the general objective is met as well as the specific objectives that consist of automating essential activities of attendance control; It has been possible to satisfactorily apply the agile extreme programming methodology, achieving the development of user stories, then the engineering tasks were successfully carried out, this allowed the user to carry out the acceptance tests that allow proceeding with the development and execution of the application of the pre-university center of the National University of San Cristóbal de Huamanga.

Keywords: extreme programming, pre-university center, attendance control

Índice

| | |
|----------|----|
| Resumen | iv |
| Abstract | v |

CAPÍTULO I INTRODUCCIÓN

| | | |
|--------|-----------------------------------|----|
| 1.1. | PLANTEAMIENTO DEL PROBLEMA | 12 |
| 1.2. | FORMULACIÓN DEL PROBLEMA | 13 |
| 1.2.1. | Problema general | 13 |
| 1.2.2. | Problemas específicos | 13 |
| 1.3. | JUSTIFICACIÓN DE LA INVESTIGACIÓN | 13 |
| 1.4. | LIMITACIONES DE LA INVESTIGACIÓN | 14 |
| 1.5. | OBJETIVOS | 14 |
| 1.5.1. | Objetivo general | 14 |
| 1.5.2. | Objetivos específicos | 14 |

CAPÍTULO II MARCO TEÓRICO

| | | |
|---------|--|----|
| 2.1. | ANTECEDENTES DE LA INVESTIGACIÓN | 15 |
| 2.2. | MARCO CONCEPTUAL | 16 |
| 2.2.1. | Control | 16 |
| 2.2.2. | Absentismo | 17 |
| 2.3. | MARCO REFERENCIAL | 17 |
| 2.3.1. | Metodología ágil | 17 |
| 2.3.2. | Metodología programación extrema | 17 |
| 2.3.3. | Artefactos de la programación extrema | 18 |
| 2.3.4. | Roles y responsabilidades | 19 |
| 2.3.5. | Fases de la metodología programación extrema | 20 |
| 2.3.6. | Programación orientada a objetos | 22 |
| 2.3.7. | Sistema de gestión de base de datos | 22 |
| 2.3.8. | Aplicación web | 23 |
| 2.3.9. | Patrón de arquitectura modelo, vista y controlador | 23 |
| 2.3.10. | REST | 24 |
| 2.3.11 | API | 24 |

**CAPÍTULO III
MATERIALES Y MÉTODOS**

| | | |
|-------------|---|-----------|
| 3.1. | TIPO Y NIVEL DE INVESTIGACIÓN | 25 |
| 3.1.1. | Tipo de investigación | 25 |
| 3.1.2. | Nivel de investigación | 26 |
| 3.2. | DISEÑO DE LA INVESTIGACIÓN | 26 |
| 3.3. | VARIABLES | 26 |
| 3.3.1. | Definición conceptual de variables | 26 |
| 3.3.2. | Definición operacional de variables | 27 |
| 3.4. | POBLACIÓN Y MUESTRA | 27 |
| 3.4.1. | Población | 27 |
| 3.4.2. | Muestra | 27 |
| 3.5. | OPERACIONALIZACIÓN DE VARIABLES | 27 |
| 3.6. | TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN | 29 |
| 3.6.1. | Técnicas | 29 |
| 3.6.2. | Instrumentos | 29 |
| 3.7. | PROCEDIMIENTOS | 29 |
| 3.7.1. | Técnicas de procesamiento de datos | 29 |
| 3.7.2. | Herramientas para el procesamiento de datos | 33 |
| 3.7.3. | Técnicas de análisis de datos | 34 |

**CAPÍTULO IV
RESULTADOS Y DISCUSIÓN**

| | | |
|-------------|------------------------------|-----------|
| 4.1. | RESULTADOS | 35 |
| 4.1.1. | Fase de exploración | 35 |
| 4.1.1.1. | Historias de usuario | 35 |
| 4.1.1.2. | Arquitectura Técnica inicial | 37 |
| 4.1.1.3. | Plan de alto nivel | 37 |
| 4.1.2. | Fase de planificación | 38 |
| 4.1.2.1. | Historia de usuario | 38 |
| 4.1.2.2. | Plan de versiones | 43 |
| 4.1.3. | Fase de iteración | 44 |
| 4.1.3.1. | Arquitectura técnica final | 44 |

| | | |
|------------------------|---|----|
| 4.1.3.2. | Tareas de ingeniería | 46 |
| 4.1.3.3. | Plan de iteración | 52 |
| 4.1.3.4. | Interfaz gráfica de usuario | 54 |
| 4.1.3.5. | Tarjeta CRC | 63 |
| 4.1.3.6. | Código Fuente | 64 |
| 4.1.3.7. | Pruebas unitarias | 76 |
| 4.1.3.8. | Pruebas de aceptación | 76 |
| 4.1.3.9. | Resumen de pruebas de aceptación | 82 |
| CAPÍTULO V | | |
| CONCLUSIONES | | |
| 5.1. | CONCLUSIONES | 85 |
| CAPÍTULO VI | | |
| RECOMENDACIONES | | |
| 6.1. | RECOMENDACIONES | 86 |
| | Referencias bibliográficas | 87 |
| | Anexos | 89 |
| | Anexo 1. Matriz de consistencia | 89 |
| | Anexo 2. Modelo de Base de Datos Física | 91 |
| | Anexo 3. Guía de Entrevista | 92 |

Lista de tablas

| | |
|--|----|
| Tabla 1. Operacionalización de la variable | 28 |
| Tabla 2. Fase de exploración | 30 |
| Tabla 3. Fase de Planeación | 30 |
| Tabla 4. Fase de Iteración | 31 |
| Tabla 5. Herramientas tecnológicas para el procesamiento de datos | 33 |
| Tabla 6. Historias de Usuario | 35 |
| Tabla 7. Plan de alto nivel | 37 |
| Tabla 8. H.U. Gestionar la cuenta del usuario | 38 |
| Tabla 9. H.U. Iniciar sesión | 39 |
| Tabla 10. H.U. Gestionar asistencia diaria en la entrada del local de clases | 39 |
| Tabla 11. H.U. Gestionar asistencia diaria en la salida del local de clases. | 40 |
| Tabla 12. H.U. Gestionar asistencia al ingresar al local el día del examen | 40 |
| Tabla 13. H.U. Gestionar asistencia en el aula donde rendirá el examen | 41 |
| Tabla 14. H.U. Generar reporte de registro de operadores | 41 |
| Tabla 15. H.U. Generar reporte de registro de auxiliar | 42 |
| Tabla 16. H.U. Generar reporte de las asistencias diarias | 42 |
| Tabla 17. H.U. Generar reporte de las asistencias al examen | 43 |
| Tabla 18. Plan de versiones | 43 |
| Tabla 19. T.I. Crear cuenta de usuario | 46 |
| Tabla 20. T.I. Iniciar sesión | 46 |
| Tabla 21. T.I. Control de ingreso a clases | 47 |
| Tabla 22. T.I. Control de salida de clases | 48 |
| Tabla 23. T.I. Control de ingreso al examen | 48 |
| Tabla 24. T.I. Control de aula en el examen | 49 |
| Tabla 25. T.I. Reporte de aulas designadas al operador | 50 |
| Tabla 26. T.I. Reporte de aulas designadas al auxiliar | 50 |
| Tabla 27. T.I. Reporte general de la asistencia en clases | 51 |
| Tabla 28. T.I. Reporte general de asistencia en el día del examen | 51 |
| Tabla 29. P.I. Primera iteración | 52 |
| Tabla 30. P.I. Segunda iteración | 52 |
| Tabla 31. P.I. Tercera iteración | 53 |
| Tabla 32. P.I. Cuarta iteración | 53 |
| Tabla 33. P.I. Quinta iteración | 53 |
| Tabla 34. Plan de iteración clasificado en fechas | 53 |
| Tabla 35. Clase Usuario | 63 |
| Tabla 36. Clase Asistencia | 63 |
| Tabla 37. Clase Persona | 63 |
| Tabla 38. Clase Aula | 64 |
| Tabla 39. P.I. Primera iteración | 76 |
| Tabla 40. P. A. Crear cuenta de usuario | 76 |
| Tabla 41. P. A. Iniciar sesión | 77 |
| Tabla 42. P. A. Control de ingreso a clases | 78 |
| Tabla 43. P. A. Control de salida de clases | 78 |
| Tabla 44. P. A. Control de ingreso al examen | 79 |
| Tabla 45. P.A. Control de aula en el examen | 79 |
| Tabla 46. P.A. Reporte de aulas designadas al operad | 80 |
| Tabla 47. P.A. Reporte de aulas designadas al auxiliar | 80 |

| | |
|---|----|
| Tabla 48. P.A. Reporte general de la asistencia en clases | 81 |
| Tabla 49. P.A. Reporte general de asistencia en el día del examen | 81 |
| Tabla 50. Resumen de pruebas de aceptación | 82 |
| Tabla 51. Reporte de pruebas de aceptación | 83 |

Lista de figuras

| | |
|--|----|
| Figura 1. Ciclo de vida de la Programación Extrema | 21 |
| Figura 2. Arquitectura técnica inicial del proyecto | 37 |
| Figura 3. Arquitectura técnica final del proyecto | 45 |
| Figura 4. Prototipo iniciar sesión | 54 |
| Figura 5. Pestaña de asistencia de clases | 55 |
| Figura 6. Registro exitoso de asistencia de clases | 55 |
| Figura 7. Alerta de registro en la asistencia a clases | 56 |
| Figura 8. Pestaña de registro en aula para la salida de clases | 56 |
| Figura 9. Registro exitoso en aula para la salida de clases | 57 |
| Figura 10. Alerta de registro en aula para la salida de clases | 57 |
| Figura 11. Reporte del auxiliar de asistencia a clases | 58 |
| Figura 12. Asistencia a examen | 58 |
| Figura 13. Registro exitoso de ingreso examen | 59 |
| Figura 14. Alerta de registro en el ingreso al examen | 59 |
| Figura 15. Pestaña de registro de asistencia al aula en el examen | 60 |
| Figura 16. Registro exitoso en el aula del examen | 60 |
| Figura 17. Alerta en el registro de asistencia en el aula del examen | 61 |
| Figura 18. Reporte de asistentes al examen del operador | 61 |
| Figura 19. Reporte del administrador del aula de clases | 62 |
| Figura 20. Reporte del administrador de las aulas del examen | 62 |
| Figura 21. Código de Spring Boot de la clase model - Aula | 64 |
| Figura 22. Código de Spring Boot de la clase model - Asistencia | 65 |
| Figura 23. Código de Spring Boot de la clase model - Estudiante | 66 |
| Figura 24. Código de Spring Boot de la clase service - AsistenciaServiceImpl | 66 |
| Figura 25. Código de Spring Boot de la clase service - AulaExamenServiceImpl | 68 |
| Figura 26. Código de Spring Boot de la clase service - AuxiliarServiceImpl | 69 |
| Figura 27. Código de Spring Boot de la clase controller - AulaController | 69 |
| Figura 28. Código de Spring Boot de la clase controller - AuxiliarController | 71 |
| Figura 29. Código Angular – auxiliar.service.ts | 72 |
| Figura 30. Código Angular – asistencia.service.ts | 72 |
| Figura 31. Código Angular componente – Registrar asistencia | 74 |
| Figura 32. Código Angular componente – Registrar de salida a clases | 75 |

CAPÍTULO I

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

El Centro preuniversitario PRE-UNSCH cuenta con un proceso del registro de control manual para la asistencia de los estudiantes que es realizado por el auxiliar asignado al local, para realizar esta función el auxiliar solicita al estudiante la ficha impresa de inscripción al momento de ingreso del estudiante al local de clases, para que posteriormente lo registre de forma manual en el libro de asistencia una tarea engorrosa que realiza diariamente por cada ciclo académico, esto no favorece al centro preuniversitario porque no existe un uso adecuado de tiempo del personal porque que abandona algunas funciones prioritarias que realiza para llevar a cabo el registro de estudiante en el libro de asistencia; cabe destacar que el ingreso de estudiantes es masivo y se vuelve incontrolable por la hora de llegada, como también no existe un reporte de los estudiantes que no asisten en tiempo real.

El centro preuniversitario ofrece vacantes de ingreso directo a la Universidad Nacional de San Cristóbal de Huamanga para los primeros puestos de cada escuela de formación profesional por tal motivo en cada ciclo académico se realiza tres exámenes para obtener el orden de mérito por escuela de los estudiantes; los cuales están distribuidos por sede principal y las distintas filiales; pero existe un gran problema en cada examen realizado porque tiene un proceso de registro de control manual de asistencia en las aulas que fueron distribuidos a los diferentes estudiantes para el examen no teniendo un filtro óptimo de quien ingresa a los locales y en algunos casos verificando la ficha impresa de inscripción manualmente al momento del ingreso a los locales del proceso de examen del centro preuniversitario, en algunos casos teniendo alumnos confusos porque ingresan a locales que no corresponden, inexistencia de un registro de alumnos que no

asisten; proceso que no favorece un óptimo uso del tiempo en el momento del ingreso del estudiante que rendirá su examen y destacando a la vez la gran asistencia masiva del estudiante al examen problema que se repite en cada ciclo académico.

En la actualidad no existe ningún sistema que permita agilizar estas actividades por tal motivo se propuso la implementación de un sistema web que gestione la asistencia de los alumnos al momento de ingresar a su aula de clases, como también el registro de asistencia de los exámenes que se desarrollan en la ciudad universitaria en los distintos pabellones, como también en las distintas sedes para el mejor control y filtro de ingreso el cual permitirá a las autoridades del centro preuniversitario realizar la supervisión de la asistencia de los estudiantes al momento de ingresar como un control interno para un mejor despliegue y coordinación en los días que se desarrolla los exámenes que son de vital importancia para la institución.

1.2. FORMULACIÓN DEL PROBLEMA

1.2.1. Problema general

¿Cómo automatizar el control de asistencia de los Estudiantes en el Centro Preuniversitario de la UNSCH, Región Ayacucho, 2023?

1.2.2. Problemas específicos

- I. ¿De qué manera automatizar el control de asistencia en aula?
- II. ¿De qué manera automatizar el control de asistencia a exámenes?

1.3. JUSTIFICACIÓN DE LA INVESTIGACIÓN

Se desea lograr la automatización del control de asistencia que permita crear mecanismos para reducir los tiempos en los procesos que no favorecen a los estudiantes al momento de ingresar a los locales, así beneficiando a los estudiantes y al personal que labora en el centro preuniversitario.

El propósito es desarrollar un software para la automatización del control de asistencia a clases permitiendo al personal que trabaja en el centro preuniversitario realizar actividades primordiales de la institución como poder realizar registros de asistencia a la hora del ingreso y salida en las clases académicas que se desarrollan diariamente como también el reporte de asistencia en tiempo real de los alumnos asistentes con la finalidad de un mejor tratamiento de datos con los

recursos tecnológicos que existen de la actualidad.

Se propone también la integración a la aplicación web para automatizar el control de asistencia al examen que se realizan tres veces por ciclo académico; se necesitara que un operario realice el control de la asistencia del estudiante al momento del ingreso al local y la verificación de su presencia en el aula donde rendirá su examen teniendo fidelidad de la información reflejada para que se pueda comprobar con los registros de ingreso del primer filtro donde se rendirá el examen.

Por tal motivo, se busca contribuir con un mejor control y transparencia; logrando que el personal auxiliar logre registrar a los estudiantes acortando tiempos por la gran cantidad de alumnos que se despliegan en sus actividades académicas diariamente en el centro preuniversitario mientras que el personal operario podrá verificar la veracidad de los datos como el control de la asistencia de los estudiantes que serán asignado a las distintas aulas según los días programados del examen; esto conllevara una reducción de costos y evitara la verificación de flujos críticos en el desarrollo de estas actividades; se quiere lograr una aplicación web que sea flexible, mantenible y escalable para la institución con el marco de trabajo de programación extrema.

1.4. LIMITACIONES DE LA INVESTIGACIÓN

En el estudio se consideró dos actividades primordiales que se automatizarán, no son las únicas actividades existentes del Centro preuniversitario, existen otras que no se tomaron en cuenta porque algunos están automatizados pero no integrados entre sí; así como también la existencia de datos dispersos porque estos son software de terceros, se requiere una mayor cantidad de data para poder realizar la automatización de otros tipos de actividades a los que no se pudo acceder por tal motivo el estudio se basa en solo dos actividades importantes de la institución.

1.5. OBJETIVOS

1.5.1. Objetivo general

Desarrollar una aplicación web para el control de asistencia de los Estudiantes en el Centro Preuniversitario de la UNSCH, Región Ayacucho, 2023.

1.5.2. Objetivos específicos

- a. Explorar, planificar e iterar para automatizar el control de asistencia en aula
- b. Explorar, planificar e iterar para automatizar el control de asistencia a exámenes

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Según Hurtado y Llanos (2021) en su tesis de pregrado “Sistema de control de asistencia a estudiantes mediante carnet virtual con código QR” refiere en su investigación que actualmente algunas Instituciones de Educación Superior en Colombia no cuentan con herramientas que registren, procesen y almacenen información sobre el acceso de los estudiantes a los servicios que se les ofrece. Esto se convierte en un gran inconveniente, cuando las IES inician un proceso de certificación de calidad, debido a que el Ministerio de Educación requiere de un sistema transversal que capture información relevante sobre los programas que se ofrecen para asegurar su calidad; El método de registro es manual por tal motivo, en el francisco Universidad Distrital Distrito José de Caldas facultad tecnológica, realizaron el desarrollo y despliegue de un sistema de registro IoT multiplataforma para el registro de acceso a la escuela, seguimiento de clases y control de asistencia a través de códigos QR; el sistema muestra logra recopilar y centralizar datos, lo que ayuda a mejorar los procesos actuales al brindar información confiable, de alta calidad y en tiempo real que se puede usar para tomar decisiones administrativas y logísticas.

Según Lomas (2022) en su tesis “Sistema de Control de Asistencia (SAM)” indica que tuvo como objetivo estudiar una empresa TIC en cántabra y que los clientes eran predominantemente empresas del sector manufacturero, en muchas de estas empresas el registro de ingreso y salida de los empleados se registraban manualmente y estas empresas carecían de una forma rápida y precisa de evaluar el nivel de ausentismo entre sus empleados o de generar los informes necesarios para cumplir con los requisitos de los sistemas de gestión del tiempo laboral sin dedicar una cantidad excesiva de tiempo y el proceso de recopilar datos manualmente en papel

para así procesarlos realizando tediosas hojas de cálculo, que resultaban no sólo en pérdida de tiempo y recursos sino también una pérdida de productividad; y el resultado en este estudio fue el desarrollo de una aplicación de Android que puede controlar a los trabajadores sin importar dónde estén o qué esté sucediendo, y esta API REST adjunta consultas de datos necesarios para generar informes como la ausencia de los trabajadores o la finalización de turnos; esto simplifica enormemente el trabajo de los responsables al realizar un seguimiento del control.

Para Sulla (2022) en su tesis de pregrado “Sistema biométrico basado en aplicaciones móviles para el control de asistencia de estudiantes del Instituto Superior Tecnológico Americana del Cusco” indica que tuvo como objetivo reducir el índice de inasistencia, el correcto control para evitar la suplantación y por ultimo aumentar la permanencia de los estudiantes implementando y desplegando un sistema biométrico mediante aplicaciones móviles para monitorear la asistencia de los estudiantes y así lograr solucionar problemas concretos relacionados con el proceso de control de asistencia al automatizarlos, en el Instituto Superior Tecnológico Americana del Cusco.

Según Viera (2020) en su tesis de pregrado “Implementación de un sistema de información para reducir errores en el proceso de control de asistencias del personal de las instituciones educativas que pertenecen a la UGEL - Huamanga de la región Ayacucho” tuvo como propósito ayudar en la administración y control del personal de las instituciones educativas que pertenecen a la UGEL HUAMANGA y así reducir los errores en este proceso de automatización del control de asistencia aplicando una arquitectura flexible que respaldara las funcionalidades de del sistema; logrando un sistema de mayor eficiencia, disponibilidad, escalable y confiable en los datos que almacena para los diferentes apoyar en la correcta toma de decisiones que necesite la institución.

2.2. MARCO CONCEPTUAL

2.2.1. Control

Según Koontz, Weihrich & Cannice (2012), menciona que el control es el proceso de monitorear y ajustar las acciones de los subordinados para garantizar que los resultados se ajusten a las metas previstas.

Para Haimann (2005) menciona que el control se define como el proceso de verificar si se están cumpliendo los objetivos así también ver así se está logrando el cumplimiento de las metas por otro parte cuánto progreso se ha logrado.

2.2.2. Absentismo

El absentismo en las instituciones es uno de los temas que menos se conversa; según la RAE el absentismo es la “Abstención deliberada de acudir al lugar donde se cumple una obligación” (Real Academia Española, s.f., definición 1), en este caso los estudiantes tienen la obligación de asistir y realizar actividades académicas en el centro preuniversitario y así prepararse para lograr una vacante, pero por la falta de un control monitoreado, la inasistencia a clase se vuelve progresivo y sin justificación alguna.

2.3. MARCO REFERENCIAL

2.3.1 Metodología ágil

Según Canós, Letelier y Penadés (2003) nos indica que cada metodología ágil se creó como alternativa de las metodologías tradicionales que a diferencias de estos están orientadas a las personas y no a los procesos por este motivo las metodologías ágiles se adaptan a las particularidades de cada proyecto logrando soluciones colaborativas y de cambio anticipado y entrega constante al cliente para una mejor retroalimentación en el desarrollo logrando refinar la calidad del software y costos reducidos en su desarrollo.

Según Medinilla (2001), afirman estar descubriendo mejores métodos de desarrollo de software compartiendo sus propias experiencias y ayudando a quienes trabajan en proyectos similares se señala también que a través del trabajo conjunto se pudo comprender mejor a las personas y las interacciones con los procesos y las herramientas de software que se basa en la documentación con la colaboración del cliente para determinar sus necesidades y la respuesta rápida a los cambios con el plan original concluyendo que las metodologías ágiles de desarrollo de software han descubierto métodos más eficientes para desarrollar software a través del uso de experiencias pasadas y apoyo externo.

2.3.2 Metodología programación extrema

Según Beck (1999/2000), indica que la programación extrema es una metodología ágil de desarrollo de software que prioriza el trabajo en equipo, alienta a los desarrolladores a aprender y crecer en el menor tiempo posible y fomenta un ambiente de trabajo positivo al enfocarse en mejorar el trabajo en equipo logrando que este sea adaptable, predecible e implementación sencilla.

Según Letelier y Sánchez (2003), indica que la programación extrema XP es una metodología ágil utilizada para el desarrollo de software que se caracteriza estar preparados para cambios durante el proyecto y están orientadas a satisfacer las necesidades del cliente logrando así establecer vínculos directos del cliente y el equipo de desarrollo alcanzando un trabajo en equipo para un éxito en el desarrollo del software alcanzando un producto de alta calidad en un corto periodo de tiempo.

De acuerdo a Sommerville (2005), indica que en la Programación Extrema XP se adopta la práctica de expresar todos los requisitos en forma de escenarios los cuales son denominados historias de usuario, estas historias de usuario son ejecutadas de manera directa siguiendo una secuencia de tareas; también nos indica que una de las prácticas más comunes en esta metodología ágil de desarrollo es la ejecución de pruebas que deben realizarse cada vez que se escribe un nuevo código esperando que se ejecuten de manera exitosa, cabe señalar que hay un pequeño intervalo de tiempo entre las distintas distribuciones del sistema.

2.3.3 Artefactos de la Programación Extrema

I. Historias de usuario

Wells (2009), menciona que las historias de usuario son un tipo de documentación que describe el comportamiento del sistema que debe tener, para estimar el tiempo de desarrollo, planificación de las iteraciones como también contribuyen las pruebas de aceptación que se desarrollarán.

Según Letelier y Penades (2006), Del mismo modo indica que las historias de usuario son un conjunto de instrumentos a través de los cuales el cliente especifica las características que debe tener el software, los cuales son dinámicos y adaptables, lo que permite agregar o modificar más historias de usuarios a medida que avanza el proyecto.

II. Tareas de ingeniería

Para Ferreira (2013), define que las descripciones detalladas de las tareas a realizar en cada historia de usuario ayudan al desarrollador a sentirse más conectado con su trabajo y le brindan una mejor comprensión del código que está escribiendo.

III. Pruebas de aceptación

Para Ferreira (2013), expresa que las pruebas de aceptación también son conocidas como las

pruebas de caja negra, a veces conocidas como pruebas de cliente, ya que las realiza la persona responsable de garantizar que las historias de usuario ejecuten la funcionalidad prometida y al cliente pueda elegir el escenario correcto que acepta la prueba.

IV. Tarjetas CRC

Para Wells (2009), Las tarjetas CRC permiten el diseño de sistemas colaborativos, mientras que las tarjetas CRC individuales se utilizan para representar objetos describiendo la clase, las responsabilidades y las clases de colaboradores del objeto.

2.3.4 Roles y responsabilidades

Según Kendall y Kendall (2005), explica cómo diferentes personas pueden desempeñar diferentes roles durante el proceso de desarrollo y cómo algunas personas pueden desempeñar múltiples roles a la vez durante el proceso y también indica los siete roles disponibles que son el programador, cliente, probador, rastreador, entrenador, Consultor y Gestor o Gran jefe.

I. Programador

Según Kendall y Kendall (2005), indica que el programador tiene la habilidad para generar el código del programa y realizar refactorización, así como también pruebas unitarias del código, para lo cual el programador y el resto del equipo deben poder comunicarse y coordinar de manera efectiva.

II. Cliente

De acuerdo a Kendall y Kendall (2005), el cliente es parte del equipo porque es alguien que está encargado de realizar las historias de usuario y las pruebas funcionales porque conoce la funcionalidad del negocio y los requerimientos que necesita así logrando verificar su implementación, además selecciona aquellas que se implementaran en cada iteración enfocándose en aportar más valor al negocio.

III. Encargado de Pruebas (Tester)

Para Kendall y Kendall (2005), indica que el encargado de las pruebas brinda asistencia al cliente en la redacción de las pruebas funcionales, así como también es encargado de monitorear el progreso general controlando el progreso hasta lograr el objetivo.

IV. Encargado de seguimiento (Tracker)

Según Kendall y Kendall (2005), El responsable debe realizar una verificación entre el tiempo real dedicado a una tarea y las estimaciones realizadas además como el monitoreo del desarrollo de la iteración como también evalúa si los objetivos son alcanzables o no dadas las limitaciones actuales de recursos y tiempo.

V. Entrenador (Coach)

Según Kendall y Kendall (2005), el entrenador o coach se encarga de todo el procedimiento para guiar eficazmente al equipo en el uso de las prácticas de XP y garantizar que se siga con el proceso adecuado por tal motivo este debe tener una comprensión profunda de la metodología XP.

VI. Consultor

Según Kendall y Kendall (2005), el consultor es un miembro externo del equipo este debe dirigir al grupo hacia una solución de un problema en particular por su experiencia ayudará a que el proyecto avance.

VII. Gestor (Big Boss)

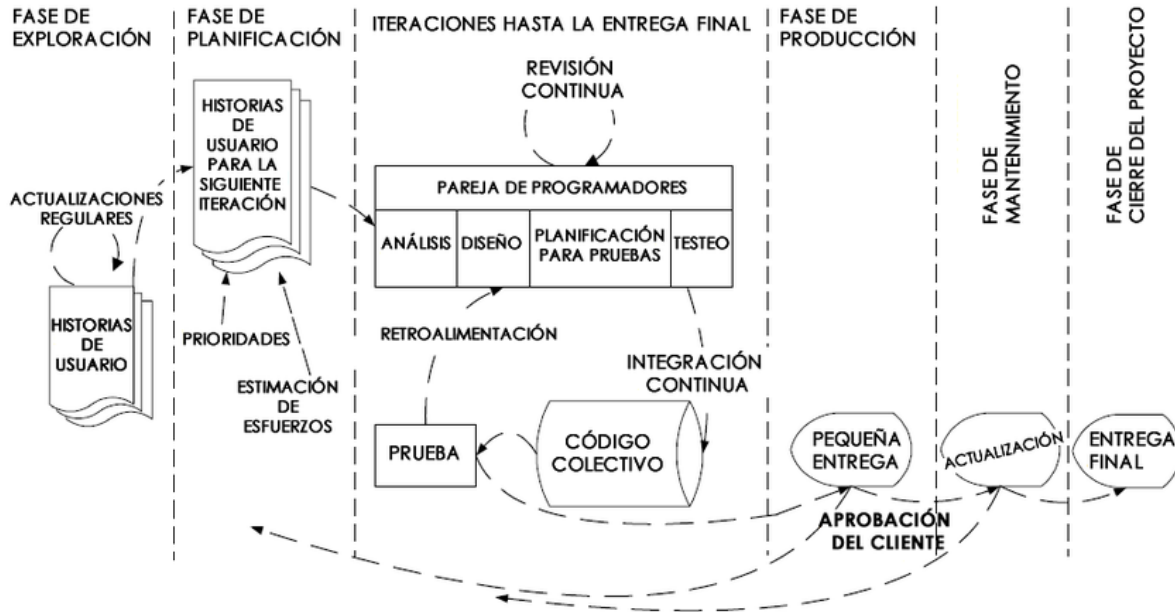
Según Kendall y Kendall (2005), indica que el gestor es quien involucra a todo el equipo y es el vínculo entre los clientes y los desarrolladores ayudando al equipo a trabajar de manera eficiente creando las condiciones adecuadas.

2.3.5 Fases de la metodología programación extrema

Para Kendall y Kendall (2005), El éxito de un proyecto XP depende de la capacidad del cliente para seleccionar el valor empresarial que se implementará en función de la capacidad del equipo para medir la funcionalidad que puede ofrecer a lo largo del tiempo ciertas acciones y actitudes determinan cómo se comportan los miembros del equipo y los clientes durante el desarrollo ágil de un proyecto; las fases de la metodología extrema consisten en los siguientes pasos a grandes rasgos:

Figura 1

Ciclo de vida de la Programación Extrema



Nota: Se muestra las fases del ciclo de vida de la programación extrema. Tomado de (Capó y Dorta, 2013)

I. Fase de exploración

Según Juskowics (2008), menciona que el alcance completo del proyecto se establece en esta etapa en relación con las historias de usuario que son escritas por el cliente y los desarrolladores evalúan y estiman el tiempo de desarrollo basándose en las en estas para poder tener cálculos preliminares en esta etapa y pueden tener cambios cuando se sometan a pruebas adicionales en iteraciones posteriores.

Para Calabria y Piriz (2003), Indica que en esta etapa los clientes proporcionan descripciones detalladas de los escenarios de las historias de usuarios mientras que el equipo de desarrollo se familiariza con las herramientas, la tecnología y las mejores prácticas que se utilizarán a lo largo del proyecto y se investigan todas las vías posibles para lograr un producto de calidad con arquitectura del sistema.

II. Fase de planificación

Según Juskowicz (2008), afirma que en la fase de planificación es donde se llegan a acuerdos entre el equipo de desarrollo y el cliente sobre el tiempo de los entregables que se realizará y el

orden en que se implementará cada una de las diferentes historias de usuario.

III. Fase de iteración

Según Joskowicz (2008), explica que en la fase de iteración se desarrollan las funcionalidades del sistema y cada iteración finaliza con algún tipo de entrega que sirve para poner en acción las historias de usuario que se asignaron a esta iteración.

IV. Fase de producción

Según Joskowicz (2008), aclara que al final de cada ciclo de la fase de producción se entregará módulos totalmente funcionales y libres de errores a pedido del cliente y este puede retrasar la puesta en producción del sistema hasta que todas sus características estén en su lugar por tal motivo indica que no se realizará más desarrollo funcional en esta etapa, sin embargo, es posible que se requieran algunos ajustes.

2.3.6 Programación orientada a objetos

Según Para Rodríguez, Sosa y Prieto (2004), indica que la Programación Orientada a Objetos es un ejemplo de desarrollo de software por su nivel de abstracción porque lograr modelar y desarrollar software complejos con gran facilidad gracias a los conceptos de clase y objeto logrando componentes más pequeños mientras que los datos se encuentran encapsulados dentro de las clases teniendo así acceso a los datos controlados e independientes, su mayor importancia consiste en la reutilización logrando la simplicidad al implementar sistemas que tienen técnicos, conceptos y herramientas que ayudan a modelar y representar con la mayor precisión posible.

Para Lowe (2005), indica que el paradigma de la Programación Orientada a Objetos es aquel en el que los objetos y sus interacciones se utilizan en el desarrollo de programas y aplicaciones informáticas estos incorporan varias metodologías, incluyendo herencia, abstracción, polimorfismo y encapsulación; los diferentes lenguajes de programación ahora ofrecen diferentes niveles de soporte para la orientación a objetos.

2.3.7 Sistema de gestión de base de datos

Según Hansen y Hansen (2004) señalan que los sistemas de gestión de bases de datos son aplicaciones altamente especializadas diseñadas para facilitar la comunicación entre una base

de datos, sus usuarios y cualquier aplicación que lo requiera.

Para Capos y Castilla (2005) menciona que uno de los objetivos principales del modelado de datos relacionales es ayudar a los usuarios a percibir y comprender las bases de datos como estructuras lógicas compuestas de conjuntos de relaciones en lugar de implementaciones físicas lo cual ayuda a lograr un alto grado de independencia de los datos también señala que un objetivo secundario del modelo es garantizar que la estructura lógica a través de la cual se ve el almacén de datos sea sencilla y coherente.

2.3.8 Aplicación web

Para Lujan (2002), define que una aplicación web es un tipo específico de aplicación cliente/servidor en la que el cliente (el navegador) y el servidor (el servidor web) están estandarizados al igual que el protocolo a través del cual se comunican (HTTP), en lugar de ser creado por el programador de la aplicación.

Según Seoane (2005), menciona que una aplicación web es un programa informático creado explícitamente para su uso en un navegador web para eso empleamos tecnologías de tres capas basadas en una arquitectura cliente-servidor: la computadora del usuario (donde se ejecutará la aplicación web) aloja la primera capa, que es responsable de cosas como mostrar datos y recuperarlos, así como generar informes y visuales; el servidor de lógica empresarial (donde se alojará la aplicación) aloja la segunda capa, que hace cosas como preparar el entorno en el que se mostrará la aplicación y en la tercera capa está alojada en el servidor de base de datos de la empresa, que es responsable de manejar las consultas enviadas desde el servidor de lógica de negocios y devolver los datos solicitados.

2.3.9 Patrón de arquitectura modelo, vista y controlador

Según Meliá (2007), define que la arquitectura de software es la descripción de los sistemas y componentes subyacentes de un sistema de software, así como las interdependencias entre ellos y generalmente se representa mediante vistas que resaltan las propiedades funcionales y no funcionales más importantes del sistema.

Según Rodríguez y Vergara (2004), indica que el patrón de diseño MVC consiste en el modelo, vista y el controlador esto se definió como una herramienta de abstracción de alto nivel para usar

en el desarrollo de la arquitectura de aplicaciones web el cual permite el desarrollo de programas de fácil mantenimiento y los beneficios son mayores a mediano y largo plazo, pero se requiere más trabajo para la planificación y codificación al dividir una aplicación en tres partes distintas, cada una de las cuales puede intercambiarse independientemente de las demás.

2.3.10 REST

Según Rosa (2018) argumenta que un servicio REST sirve para conectar varios sistemas mediante el protocolo HTTP que contiene la información necesaria el cual permite la recopilación, operación y entrega de datos desde y hacia el cliente, permitiendo separar al cliente del servidor y haciendo mucho más fácil la integración de API logrando más fácil el desarrollo de una API REST.

Para Zulian (2011), nos indica que un servicio REST es un Representational State Transfer o Transferencia de Estado Representacional los cuales están enfocados en el rendimiento del sistema, incluyendo el acceso al estado de los recursos y como se trasladan por métodos HTTP con los protocolos definidos hacia los clientes; los servicios web REST debe incluir en la petición todos los parámetros y datos necesarios para que el servidor genere una respuesta completa e independiente logrando mejorar el rendimiento de los servicios web.

2.3.11 API

Según Julián (2017) argumenta que una API es un acrónimo en inglés que apunta a la expresión Application Programming Interface cuya traducción es Interfaz de Programación de Aplicaciones además define que es un método estandarizado de comunicación entre programas informáticos esta idea se refiere a que las funciones y métodos abstractos proporcionados por una biblioteca de programación particular pueda ser utilizada por alguna otra aplicación.; en otras palabras, el usuario no percibe como las distintas plataformas pueden comunicarse entre sí porque a través de API lo realizan sin necesidad de que el usuario intervenga.

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1. TIPO Y NIVEL DE INVESTIGACIÓN

3.1.1. Tipo de investigación

Según Manterola y Otzen (2014), indica que el tipo de investigación observacional es aquel en el que el investigador sólo observa y registra los eventos a medida que ocurren, en lugar de intentar influir en ellos o alterarlos de alguna manera. Para nuestro caso el estudio es observacional porque no existe intervención a la variable control de asistencia para desarrollar la aplicación web.

Según Alvarado, Pineda y Canales (1994), definen que el estudio es prospectivo cuando se recopilan datos en tiempo real ya que se está monitoreando con antelación ya que el evento o fenómeno está programado para ocurrir. En nuestro caso es prospectivo porque se recolectan datos primarios para el desarrollo de la aplicación web.

Según Palomino (2015), sostiene que se trata de un estudio transversal cuando los datos se recopilan en un único momento en el tiempo como también se caracteriza por su objetivo de detallar variables y la relación que tienen estos en un momento determinado. El estudio es transversal, porque se hace una medida de la variable de interés para desarrollar la aplicación web.

Según Mejía, Reyes y Sánchez (2018), menciona que el tipo de estudio es descriptivo porque implica detallar el estado actual de los aspectos más cruciales del fenómeno que se estudia. En nuestro caso es descriptiva, porque tenemos una sola variable de estudio para desarrollar un prototipo de aplicación web.

3.1.2. Nivel de investigación

Supo (2014), el nivel de investigación es descriptivo cuando se utilizan el análisis para describir un objeto de estudio o una situación específica destacando sus características y cualidades más destacadas y así poder interpretarlo y combinar con otros criterios de clasificación que ayudan a ordenar, agrupar o sistematizar los objetos involucrados en la investigación.

La investigación es de nivel descriptivo y se desarrollará un prototipo de software, que es la aplicación web para control de asistencia de estudiantes del Centro Pre Universitario de la UNSCH; por esta consideración es de nivel descriptivo.

3.2. DISEÑO DE LA INVESTIGACIÓN

El diseño de la investigación es no experimental, porque no se manipula las variables y solo se observa cómo se dan en su contexto natural en el cual podremos describir, identificar y predecir el comportamiento de las variables, así como, es prospectivo-transversal. Prospectivo, porque se realiza entrevista a los usuarios del software para definir las historias de usuario, tareas de ingeniería, es transversal porque se ha realizado en un solo momento las entrevistas a todos los usuarios.

3.3. VARIABLES

3.3.1. Definición conceptual de variables

Variable de interés

Control de asistencia: El control de asistencia es el proceso para realizar un seguimiento de los horarios de llegada y salida de una organización, así como de cualquier incidente o ausencia que deba ser monitoreado y analizado para garantizar que se cumplan los objetivos.

Variables descriptivas (dimensiones)

Asistencia en aula: La asistencia regular a clase es un factor crítico en el desarrollo académico de los estudiantes ya que facilita la adquisición de nuevas habilidades, conocimientos y actitudes.

Asistencia en examen: La participación en los exámenes es un punto clave para mostrar el progreso académico de los estudiantes porque permite la autoevaluación, el desempeño y el crecimiento como también la demostración de las habilidades académicas y su responsabilidad

con sus obligaciones.

3.3.2. Definición operacional de variables

Variable de interés

Control de asistencia

Variables descriptivas (dimensiones)

Asistencia en aula: Representa a los procesos que se realizan dentro del centro preuniversitario para tener el control diario de los estudiantes.

Asistencia en examen: Representa a los procesos que se realizan dentro del centro preuniversitario para tener el control de los estudiantes en los días de examen.

3.4. POBLACIÓN Y MUESTRA

3.4.1. Población

Para Luzardo y Jiménez (2018) indican que la población es “un conjunto, finito o infinito, de seres vivos, elementos o cosas sobre las cuales están definidas características o variables que interesan analizar” (p. 41).

La población está compuesta por 3 procesos de control de asistencia en aula y 3 procesos de control de asistencia a exámenes, de estudiantes del Centro preuniversitario de la Universidad Nacional de San Cristóbal de Huamanga el año 2023.

3.4.2. Muestra

Para Hernández (2014), indica que la muestra de una población en estudio debe ser determinada para un correcto muestreo y análisis según la cantidad de muestra el que permitirá predecir correctamente los resultados.

No existe muestra, será un censo, porque se estudiará 3 procesos de control de asistencia en aula y 3 procesos de control de asistencia a exámenes.

3.5. OPERACIONALIZACIÓN DE LAS VARIABLES

La operacionalización de la variable control de asistencia, se realiza para medir la variable de

interés, mediante las variables descriptivas y los indicadores, con la finalidad de formular las preguntas o ítems, para el instrumento de recolección de datos, que permita automatizar los 3 procesos de control de asistencia en aula y 3 procesos de control de asistencia a exámenes.

Tabla 1

Operacionalización de la variable

| Variable | Dimensiones | Indicador | Ítem |
|-----------------------|----------------------|------------------|--|
| Control de asistencia | Asistencia en aula | Inasistencia | <p>¿Cómo el estudiante se registra al momento del ingreso a las instalaciones donde se impartirán las clases?</p> <p>¿Cuántas inasistencias debe tener el estudiante para llamarle la atención?</p> <p>¿Cómo toman decisiones a consecuencia de un alto porcentaje de inasistencia?</p> |
| | | Monitoreo | <p>¿Cómo verifica la inasistencia en tiempo real?</p> <p>¿Qué tipo de control general de los estudiantes existe?</p> <p>¿Cómo se monitorea la puntualidad de los estudiantes?</p> <p>¿Qué herramienta digital se necesita para facilitar el registro de ingreso?</p> |
| Asistencia en examen | Asistencia en examen | Inasistencia | <p>¿Cómo registrar la asistencia de estudiantes en los exámenes?</p> <p>¿Qué herramienta facilita el registro de asistencia en aula para el examen?</p> <p>¿De qué manera al ingresar a los locales del examen se toma la asistencia a los estudiantes?</p> <p>¿Cuándo ingresan al aula, donde se rendirá el examen, se toma asistencia?</p> |
| | | Monitoreo | <p>¿Cómo verifica la inasistencia en tiempo real a los exámenes?</p> <p>¿Cómo verifica la identidad del estudiante que rendirá el examen?</p> <p>¿Cómo verifica que el estudiante que se</p> |

encuentra en su aula, el día del examen paso por los filtros de ingreso?

¿Qué herramienta digital permite monitorear y facilitar el registro de los estudiantes el día del examen?

3.6. TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN

3.6.1. Técnicas

Para Espinoza (2014), menciona que la entrevista es una técnica para recopilar información importante de un punto clave, así como las características de un problema dado que estos datos podrían ser nuevos o como también complementos que ayudarán a cuantificar las características y la naturaleza de la investigación.

En este estudio se usa la técnica de la entrevista, que permite en entrevistar a los usuarios directos e indirectos del proceso de asistencia, para el desarrollo de la aplicación web y, recabar información para las historias de usuarios de la aplicación web del centro preuniversitario de la Universidad Nacional de San Cristóbal de Huamanga.

3.6.2. Instrumentos

En instrumento de recolección de datos es la guía de entrevista, para definir las tareas de ingeniería relacionados a las historias de usuario, como; las funciones, los requisitos y los procesos que conforma la aplicación web del centro preuniversitario de la Universidad Nacional de San Cristóbal de Huamanga.

3.7. PROCEDIMIENTOS

3.7.1. Técnicas de procesamiento de datos

Para procesar las entrevistas, que nos indican los requerimientos de la aplicación web y funcionalidad del software, se realiza mediante el proceso programación extrema, como se muestra en las siguientes tablas 2, 3 y 4:

Tabla 2*Fase de exploración*

| Tarea | Artefacto | Técnica | Responsables |
|---|------------------------------|--|--------------------------------------|
| Escribir historias de usuarios | Historia de usuario | Describe brevemente las historias de usuario y las reglas de negocio (lo que debe hacer el sistema). Desglose las grandes historias de usuarios. | Cliente |
| Probar las tecnologías a utilizar | Arquitectura técnica inicial | Explorar los usos de la tecnología. Rendimiento de la tecnología de prueba Definir la tecnología a utilizar. | Cliente Programador Entrenador |
| Estimar esfuerzo para historia de usuario | Plan de alto nivel | Anticiparse a las historias de los usuarios Esfuerzo estimado para desarrollar historias de usuario (semanas) | Programador |

Nota: Tomado de “Comparación de dos Procesos de Desarrollo de Software Usando los Métodos ICONIX y XP” (Porrás, 2010).

Tabla 3*Fase de Planeación*

| Tarea | Artefacto | Técnica | Responsables |
|-------------------------------------|---------------------------------|--|------------------------|
| Reescribir las historias de usuario | Historia de usuario | Describe brevemente las historias de usuario y la regla de negocio | Cliente |
| Formular el plan de versiones | Plan de versión (una iteración) | <ul style="list-style-type: none"> • Introducir nuevos requisitos del software. • Definir prioridad para cada historia de usuario por necesidad del negocio. | Cliente Programador |

- Utilizar técnicas de elaboración del plan de alto nivel.
- Calcule y asigne el esfuerzo (semanas) para cada historia de usuario en función del tiempo asignado para la planificación, el desarrollo, la implementación y las pruebas.
- Estimar y asignar riesgo a cada historia de usuario en función a las situaciones que afectan la estimación del esfuerzo.
- Actualizar tarjeta de historia de usuario

Fuente: Tomado de “Comparación de dos Procesos de Desarrollo de Software Usando los Métodos ICONIX y XP” (Porrás, 2010).

Tabla 4

Fase de Iteración

| Tarea | Artefacto | Técnica | Responsables |
|---------------------------------|----------------------|---|---|
| Definir la arquitectura técnica | Arquitectura técnica | <ul style="list-style-type: none"> • Actualizar la arquitectura técnica inicial. • Usar características del negocio. • Utilizar arquitectura por capas. • Integrar frameworks. | Cliente Programador Entrenador |
| Escribir tareas de ingeniería | Tarea de ingeniería | <ul style="list-style-type: none"> • Dividir cada historia de usuario en tareas, describir usando reglas del negocio cada tarea de ingeniería. • Estimar y asignar esfuerzo para desarrollar una tarea de ingeniería • Asignar una tarea de ingeniería al programador. | Cliente Programador Programador Entrenador |
| Formular el plan de | Plan de | <ul style="list-style-type: none"> • Utilizar el plan de versión | Programador |

| | | | |
|---|----------------------------------|---|---------------------------|
| iteraciones | iteraciones | <ul style="list-style-type: none"> • Actualizar el plan con tareas de ingeniería de la siguiente iteración • Actualizar el cuándo fallo prueba de aceptación • Actualizar el plan con tareas no concluidas • Actualizar las tarjetas de tarea de ingeniería | Entrenador |
| Interfaz preliminar | GUI | <ul style="list-style-type: none"> • Diseñar la interfaz preliminar relacionada a cada historia de usuario. | Cliente Programador |
| Escribir tarjetas CRC para cada tarea de ingeniería | Tarjeta CRC | <ul style="list-style-type: none"> • Diseñar para una tarea de ingeniería de forma simple • Rediseñar por falla de prueba de aceptación una tarea. • Identificar responsabilidades. • Identificar colaboración • Identificar Atributos | Cliente Programador |
| Modelar procesos | Diagrama de modelado de procesos | <ul style="list-style-type: none"> • Definir las actividades y flujo de trabajo dependiendo del requerimiento de la organización | Analista de procesos |
| Implementar código para la clase entidad | Código fuente | <ul style="list-style-type: none"> • Escribir código fuente o generar con una herramienta usando tarjetas CRC | Programador |
| Crear pruebas unitarias para las clases control | Prueba unitaria | <ul style="list-style-type: none"> • Escribir código fuente para una prueba unitaria, usando una herramienta | Programador |
| Implementar código fuente | Código fuente | <ul style="list-style-type: none"> • Desarrollar una tarea de ingeniería • Hacer refactoring • Mover desarrolladores | Programador Supervisor |
| Ejecutar pruebas unitarias | Código fuente | <ul style="list-style-type: none"> • Ejecutar el módulo de cada prueba unitaria • Modificar código fuente si la | Programador |

| | | | |
|--|-----------------------------------|--|------------------------------|
| | | prueba unitaria muestra resultado incorrecto | |
| Realizar integración continúa | Código fuente | <ul style="list-style-type: none"> • Integrar las tareas para una historia de usuario • Mantener sistema integrado todo el tiempo | Programador |
| Realizar pruebas de integración para una historia de usuario | Reporte de pruebas de integración | <ul style="list-style-type: none"> • Integrar continuamente al concluir las tareas de una historia de usuario • Verificar que las pruebas de integración pasan al 100% | Programador |
| Ejecutar pruebas de aceptación | Reporte de pruebas de aceptación | <ul style="list-style-type: none"> • Correr la última versión de una iteración • Utilizar los casos de prueba de aceptación | Cliente encargado de pruebas |

Fuente: Tomado de “Comparación de dos Procesos de Desarrollo de Software Usando los Métodos ICONIX y XP” (Porrás, 2010).

3.7.2. Herramientas para el procesamiento de datos

A continuación, en la tabla 5, se muestra las herramientas de procesamiento de datos utilizadas a lo largo de este proceso de desarrollo.

Tabla 5

Herramientas tecnológicas para el procesamiento de datos

| Software | Fabricante | Servicio |
|-----------------------|--------------------|--|
| Windows 11 | Microsoft | Sistema operativo moderno, seguro y optimizado para el trabajo ofrece una nueva interfaz de usuario, características mejoradas y una mejor experiencia de usuario en general. |
| Microsoft Office 2016 | Microsoft | Microsoft Office es un conjunto de aplicaciones de productividad que permite a las personas crear, editar, compartir y colaborar en documentos, hojas de cálculo, presentaciones, correos electrónicos, etc. |
| Mysql | Oracle Corporation | MySQL es un sistema de gestión de bases de datos |

| | | |
|--------------------|-------------------|--|
| | | relacionales de código abierto, rápido, probado y escalable |
| Node Js | Ryan Dahl | NodeJs brinda servicio de código abierto y es un framework de JavaScript que es multiplataforma y sirve para crear aplicaciones web dinámicas, escalables y de alta calidad. |
| Postman | Eula | Herramienta para hacer peticiones a APIs y generar colecciones de peticiones que nos permitan probarlas de una manera rápida y sencilla. |
| Visual Studio Code | MIT License | Visual Studio Code es un editor de código abierto desarrollado por Microsoft el cual proporciona un editor ligero y altamente extensible que se puede utilizar para desarrollar teniendo funciones como resaltado de sintaxis, finalización de código inteligente, fragmentos y refactorización de código, así como depuración y supervisión integrada de Git. |
| Angular | Google | Los servicios de Angular son orientados a componentes y estos son objetos que proporcionan datos o funcionalidad a otros componentes tiene como funcionalidad inyectar en los componentes a través de la inyección de dependencias así logrando desarrollar aplicaciones web robustas. |
| Spring Boot | Pivotal Software. | Spring Boot es un espacio de trabajo que facilita el desarrollo de aplicaciones Java para poder proporcionar una serie de características que hacen que el desarrollo de aplicaciones sea más rápido y fácil, como la autoconfiguración, la inyección de dependencias y el soporte para el desarrollo web. |

3.7.3. Técnicas de análisis de datos

Los datos obtenidos mediante la guía de entrevista, se analizan mediante pequeños ciclos de desarrollo, prueba continuas e interacción con el cliente, logrando identificar patrones y tendencias que son útiles para la toma de decisiones en la aplicación web para el Centro Preuniversitario de la UNSCH.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Fase de exploración

De acuerdo con la metodología de programación extrema tenemos la fase de exploración que debe reflejar las historias de usuarios generando también la arquitectura temprana de los artefactos necesarios para modelar e implementar el software del centro preuniversitario de la Universidad Nacional de San Cristóbal de Huamanga.

4.1.1.1. Historias de usuario

En este apartado enumeramos las historias de usuarios que caracterizan los distintos puntos clave del entregable del proyecto.

Tabla 6

Historias de Usuario

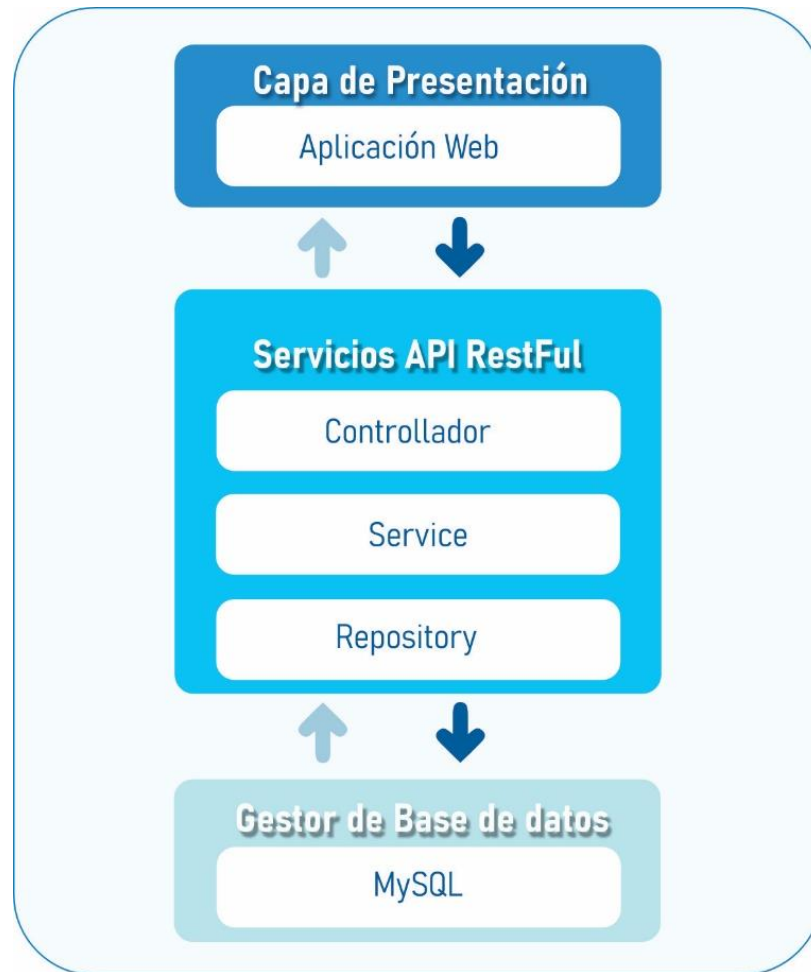
| N° | Historias de usuario | Descripción |
|----|---------------------------------|---|
| 1 | Gestionar la cuenta del usuario | Permitirá al administrador acceder a la aplicación para que pueda otorgar cuentas de usuarios y permisos necesarios que estarán definidos por los siguientes roles; administrador, auxiliar y operador. |
| 2 | Gestionar sesión | Permite el acceso a los usuarios de la aplicación web por medio de la autenticación de usuario y contraseña que le ha sido designado por el administrador según su rol de usuario. |

| | | |
|----|---|--|
| 3 | Gestionar asistencia diaria en la entrada del local de clases | Permite a los auxiliares registrar el ingreso de los estudiantes al momento de su ingreso a los locales donde se les impartirá clases presenciales |
| 4 | Gestionar asistencia diaria en la salida del local de clases | Permite a los auxiliares registrar la salida de los estudiantes cuando estos se retiren de los locales donde se les imparten las clases presenciales |
| 5 | Gestionar asistencia al ingresar al local el día del examen | Permite a los operadores registrar el ingreso de los estudiantes al momento de ingresar a los locales de exámenes donde rendirán su examen. |
| 6 | Gestionar asistencia en el aula donde rendirá el examen | Permite a los operadores volver a registrar la asistencia de los estudiantes en sus respectivas aulas designadas para rendir el examen. |
| 7 | Generar reporte de registro de operadores | Los operadores podrán visualizar el registro de entrada como el registro en aula de los estudiantes designados para su local en el día del examen. |
| 8 | Generar reporte de registro de auxiliar | Los auxiliares podrán visualizar el registro de entrada como el registro en salida de los estudiantes designados para su local en y sus respectivas aulas. |
| 9 | Generar reporte de las asistencias diarias | El Administrador debe ser capaz de monitorear la asistencia de todas las aulas asignadas a los estudiantes que asisten a clases. |
| 10 | Generar reporte de las asistencias al examen | El Administrador debe ser capaz de monitorear la asistencia de todas las aulas asignadas a los estudiantes que asisten el día del examen. |

4.1.1.2. Arquitectura Técnica inicial

Figura 2

Arquitectura técnica inicial del proyecto



4.1.1.3. Plan de alto nivel

Tabla 7.

Plan de alto nivel

| N° | Historias de usuario | Esfuerzo (Días) |
|----|---------------------------------|-----------------|
| 1 | Gestionar la cuenta del usuario | 3 |
| 2 | Gestionar sesión | 5 |

| | | |
|----|---|---|
| 3 | Gestionar asistencia diaria en la entrada del local de clases | 4 |
| 4 | Gestionar asistencia diaria en la salida del local de clases | 4 |
| 5 | Gestionar asistencia al ingresar al local el día del examen | 3 |
| 6 | Gestionar asistencia en el aula donde rendirá el examen | 3 |
| 7 | Generar reporte de registro de operadores | 6 |
| 8 | Generar reporte de registro de auxiliar | 6 |
| 9 | Generar reporte de las asistencias diarias | 8 |
| 10 | Generar reporte de las asistencias al examen | 8 |

4.1.2. Fase de planificación

4.1.2.1. Historia de usuario

Tabla 8.

H.U. Gestionar la cuenta del usuario

| Historia de usuario | |
|---|------------------------------------|
| Número: 1 | Usuario: Administrador |
| Nombre de la historia: Gestionar la cuenta del usuario | |
| Prioridad del negocio: Alto | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: Permitirá al administrador acceder a la aplicación para que pueda otorgar cuentas de usuarios y permisos necesarios que estarán definidos por los siguientes roles; administrador, auxiliar y operador de examen. | |
| Observaciones: Ninguno | |

Tabla 9.

H.U. Iniciar sesión

| Historia de usuario | |
|--|------------------------------------|
| Número: 2 | Usuario: Todos |
| Nombre de la historia: Gestionar sesión | |
| Prioridad del negocio: Alto | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: Permite el acceso a los usuarios de la aplicación web por medio de la autenticación de usuario y contraseña que le ha sido designado por el administrador según su rol de usuario. | |
| Observaciones: Ninguno | |

Tabla 10.

H.U. Gestionar asistencia diaria en la entrada del local de clases

| Historia de usuario | |
|--|-----------------------------------|
| Número: 3 | Usuario: Auxiliar |
| Nombre de la historia: Gestionar asistencia diaria en la entrada del local de clases | |
| Prioridad del negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El auxiliar será capaz de registrar la asistencia de los alumnos al momento de su ingreso al local con un lector de barras y el DNI que otorga el alumno con la finalidad de tener el control de asistencia registrado en el sistema de los alumnos que acuden a clases diariamente en los diferentes locales. | |
| Observaciones: Ninguno | |

Tabla 11.

H.U. Gestionar asistencia diaria en la salida del local de clases.

| Historia de usuario | |
|--|------------------------------------|
| Número: 4 | Usuario: Auxiliar |
| Nombre de la historia: Gestionar asistencia diaria en la salida del local de clases | |
| Prioridad del negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El auxiliar será capaz de registrar la asistencia de salida de los alumnos minutos antes de la hora de salida el cual deberá ingresar a las diferentes aulas del centro preuniversitario con el lector de código de barras y el alumno debe entregar su DNI con la finalidad de tener un control adecuado de la asistencia de alumnos que asisten diariamente. | |
| Observaciones: Ninguno | |

Tabla 12.

H.U. Gestionar asistencia al ingresar al local el día del examen

| Historia de usuario | |
|--|-----------------------------------|
| Número: 5 | Usuario: Operador |
| Nombre de la historia: Gestionar asistencia al ingresar al local el día del examen | |
| Prioridad del negocio: Alta | Riesgo en desarrollo: Alta |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El operador será capaz de registrar la asistencia de los alumnos al momento de su ingreso al local donde el alumno rendirá el examen con un lector de barras y el DNI que otorga el alumno, podrá visualizar los datos completos del alumno, el aula donde rendirá su examen y la foto que subió en el momento de su matrícula con la finalidad de tener el control del sistema de asistencia de alumnos que acuden al examen en los diferentes locales. | |
| Observaciones: Si el alumno no pertenece al local, este no debe registrar e indicar con un | |

dónde debe rendir su examen

Tabla 13.

H.U. Gestionar asistencia en el aula donde rendirá el examen

| Historia de usuario | |
|--|------------------------------------|
| Número: 6 | Usuario: Operador |
| Nombre de la historia: Gestionar asistencia en el aula donde rendirá el examen. | |
| Prioridad del negocio: Alta | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El operador será capaz de registrar la asistencia de los alumnos al momento de su permanencia en el aula donde el alumno rendirá el examen mediante un lector de barras y el DNI que otorga el alumno, podrá visualizar los datos completos del alumno, el aula donde rendirá su examen y la foto que subió en el momento de su matrícula con la finalidad de tener verificar el primer filtro de ingreso en los diferentes locales. | |
| Observaciones: Si el alumno no se registró en el ingreso, no debería registrar en el aula | |

Tabla 14.

H.U. Generar reporte de registro de operadores

| Historia de usuario | |
|---|------------------------------------|
| Número: 7 | Usuario: Operador |
| Nombre de la historia: Generar reporte de registro de operadores | |
| Prioridad del negocio: Media | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El Operador será capaz de monitorear la asistencia de los alumnos al momento de su ingreso al local y su permanencia en el aula asignada en el inicio, durante y al finalizar del examen con la finalidad de tener un mejor control con el sistema de gestión asistencia de | |

los alumnos en el día del examen.

Observaciones: Ninguno

Tabla 15.

H.U. Generar reporte de registro de auxiliar

| Historia de usuario | |
|---|------------------------------------|
| Número: 8 | Usuario: Auxiliar |
| Nombre de la historia: Generar reporte de registro de auxiliar | |
| Prioridad del negocio: Media | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El auxiliar será capaz de monitorear la asistencia de los alumnos al momento de su ingreso al local y su permanencia en el aula de clases en el inicio, durante y al finalizar la labor académica con la finalidad de tener un mejor control con el sistema de gestión asistencia de los alumnos diariamente. | |
| Observaciones: Ninguno | |

Tabla 16.

H.U. Generar reporte de las asistencias diarias

| Historia de usuario | |
|---|------------------------------------|
| Número: 9 | Usuario: Administrador |
| Nombre de la historia: Generar reporte de las asistencias diarias | |
| Prioridad del negocio: Media | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El Administrador será capaz de visualizar el reporte general de la asistencia de los alumnos de todas los locales y aulas con la finalidad de tener la información a la mano y la | |

capacidad de tomar decisiones según los objetivos institucionales.

Observaciones: Ninguno

Tabla 17.

H.U. Generar reporte de las asistencias al examen

| Historia de usuario | |
|--|------------------------------------|
| Número: 10 | Usuario: Administrador |
| Nombre de la historia: Generar reporte de las asistencias al examen | |
| Prioridad del negocio: Media | Riesgo en desarrollo: Media |
| Puntos estimados: 1 | Iteración asignada: 1 |
| Descripción: El Administrador será capaz de monitorear y vitalizar la asistencia de los alumnos al momento de su ingreso al local y permanencia en el aula asignada en el inicio, durante y al finalizar el examen con la finalidad de tener la información a la mano y la capacidad de tomar decisiones según los objetivos institucionales. | |
| Observaciones: Ninguno | |

4.1.2.2. Plan de versiones

Tabla 18

Plan de versiones

| N° | Historia de usuario | Tiempo Estimado | | | Iteración |
|----|---|-----------------|--------|----------------|-----------|
| | | Prioridad | Riesgo | Días estimados | |
| 1 | Gestionar la cuenta del usuario | Alta | Media | 3 | 1 |
| 2 | Gestionar sesión | Alta | Media | 5 | 1 |
| 3 | Gestionar asistencia diaria en la entrada del local de clases | Alta | Media | 4 | 2 |
| 4 | Gestionar asistencia diaria en la salida del local de clases | Alta | Alta | 4 | 2 |

| | | | | | |
|-----------|---|-------|-------|---|---|
| 5 | Gestionar asistencia al ingresar al local el día del examen | Alta | Media | 3 | 3 |
| 6 | Gestionar asistencia en el aula donde rendirá el examen | Alta | Alta | 3 | 3 |
| 7 | Generar reporte de registro de operadores | Media | Media | 6 | 4 |
| 8 | Generar reporte de registro de auxiliar | Media | Media | 6 | 4 |
| 9 | Generar reporte de las asistencias diarias | Media | Media | 8 | 5 |
| 10 | Generar reporte de las asistencias al examen | Media | Media | 8 | 5 |

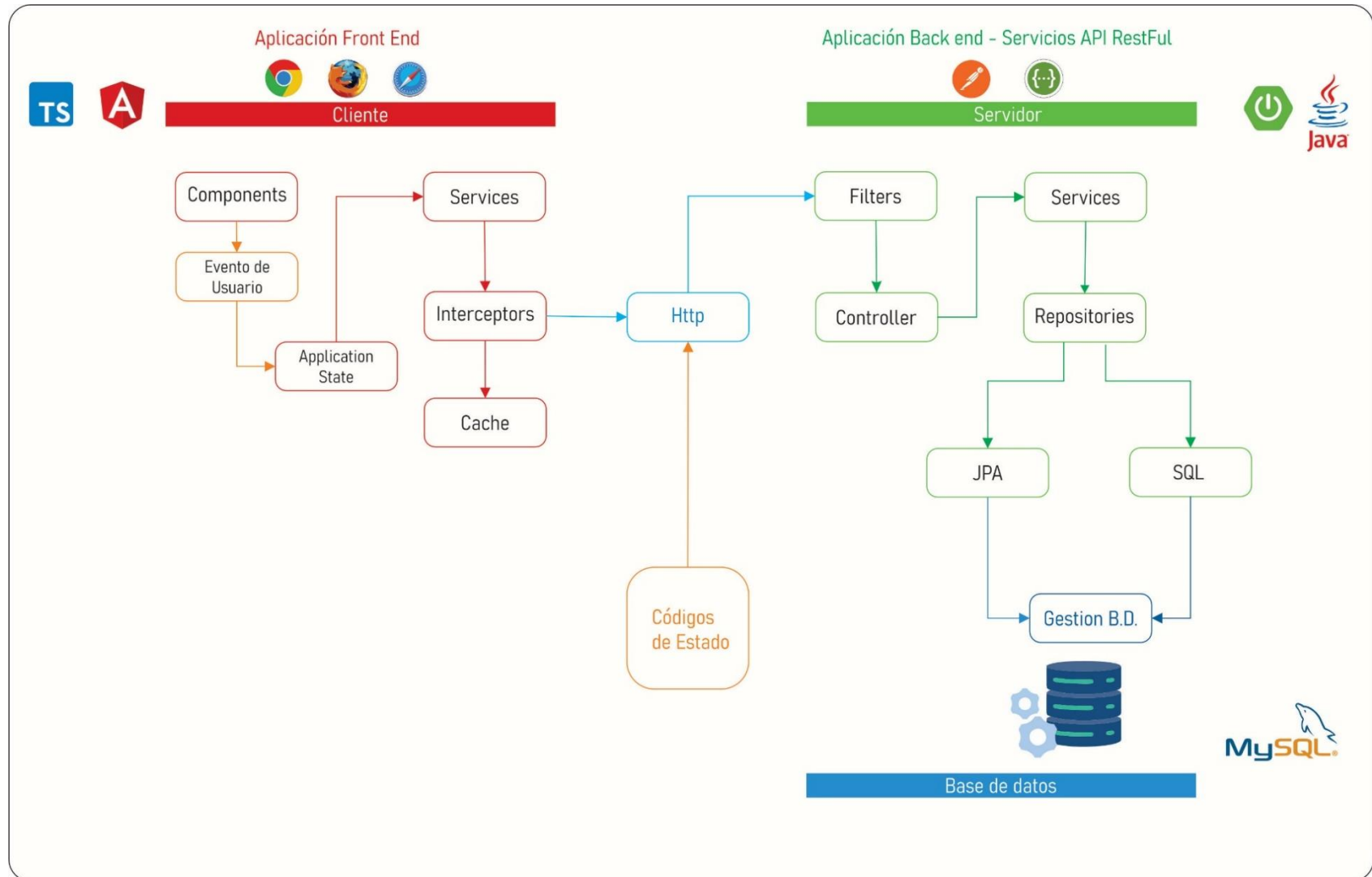
4.1.3. Fase de iteración

Según el marco de referencia citado en el Capítulo II sobre Programación Extrema, los artefactos, la arquitectura técnica, las tareas de ingeniería, los planes de iteración, los casos de prueba de aceptación, las interfaces gráficas de usuario, las tarjetas (Clase-Responsabilidad-Cooperación), una base de datos física, la clase entidad, El código fuente de clase, el código de prueba unitaria, el informe de prueba unitaria y el proyecto representativo se adquieren en la fase de iteración.

4.1.3.1. Arquitectura técnica final

Figura 3

Arquitectura técnica final del proyecto



4.1.3.2. Tareas de ingeniería

Cabe precisar que cada historia de usuario descrita anteriormente se divide en tareas, estas también se les conoce como tareas de ingeniería las cuales facilitarán la implementación del proyecto; por tal motivo mostraremos a continuación las tareas de ingeniería del proyecto del centro preuniversitario de la Universidad Nacional de san Cristóbal de Huamanga.

Tabla 19.

T.I. Crear cuenta de usuario

| Tarea de ingeniería | |
|---|---|
| Número de tarea de ingeniería: 1 | Número de historia de usuario: 1 |
| Nombre de la tarea: Gestionar la cuenta del usuario | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 25/05/2023 | Fecha fin: 28/05/2023 |
| Programador responsable: Tueros Puchuri, Elmer | |
| Descripción: El administrador accede al sistema, el sistema muestra la interface “crear nuevo usuario” e ingresa el nombre de usuario y la contraseña asignándole un rol, luego hace clic en el botón crear usuario, el sistema le muestra un mensaje con el mensaje de “operación satisfactoria”. Si los datos son incorrectos el sistema muestra un mensaje de error. | |
| Observación: El administrador podrá crear usuarios también por servicios API REST | |

Tabla 20.

T.I. Iniciar sesión

| Tarea de ingeniería | |
|--|---|
| Número de tarea de ingeniería: 2 | Número de historia de usuario: 1 |
| Nombre de la tarea: Gestión de sesión | |

Tipo de tarea: Desarrollo

Puntos estimados: 1

Fecha inicio: 28/05/2023

Fecha fin: 30/05/2023

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El usuario accede al sistema, el sistema muestra la interface "Inicio de sesión" e ingresa el nombre de usuario y la contraseña, luego hace clic en el botón de Ingresar, el sistema le muestra la pantalla principal. Si los datos son incorrectos el sistema muestra un mensaje de error.

Observación: Ninguno

Tabla 21.

T.I. – Control de ingreso a clases

Tarea de ingeniería

Número de tarea de ingeniería: 3

Número de historia de usuario: 1

Nombre de la tarea: Gestionar asistencia diaria en la entrada del local de clases

Tipo de tarea: Desarrollo

Puntos estimados: 1

Fecha inicio: 01/06/2022

Fecha fin: 04/06/2022

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El operador accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña registrar asistencia de entrada, el sistema muestra la interface "Registro de ingreso" y con un lector de barras realizará la lectura del código de barras del DNI el cual facilita el estudiante al momento de ingresar al local, luego el sistema busca y registra la fecha y hora del DNI ingresado, Seguidamente el sistema muestra en la pantalla el aula a cuál corresponde y los datos del alumno. Si los datos son incorrectos el sistema muestra un mensaje de error.

Observación: Ninguno

Tabla 22.*T.I. – Control de salida de clases*

| Tarea de ingeniería | |
|---|---|
| Número de tarea de ingeniería: 4 | Número de historia de usuario: 1 |
| Nombre de la tarea: Gestionar asistencia diaria en la salida del local de clases | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 05/06/2023 | Fecha fin: 08/06/2023 |
| Programador responsable: Tueros Puchuri, Elmer | |
| Descripción: | |
| <p>El operador accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña “registrar asistencia de salida”, el sistema muestra la interface “Registro de salida” y con un lector de código barras realizará la lectura del código de barras del DNI el cual facilita el estudiante al momento que el auxiliar vea por conveniente para la salida del local, luego el sistema busca y registra la fecha y hora del DNI ingresado, Seguidamente el sistema muestra en la pantalla con los datos del alumno. Si los datos son incorrectos el sistema muestra un mensaje de error.</p> | |
| Observación: El alumno no podrá registrar su salida si no registro su ingreso | |

Tabla 23.*T.I. – Control de ingreso al examen*

| Tarea de ingeniería | |
|--|---|
| Número de tarea de ingeniería: 5 | Número de historia de usuario: 1 |
| Nombre de la tarea: Gestionar asistencia al ingresar al local el día del examen | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 09/06/2023 | Fecha fin: 12/06/2023 |

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El operador accede al sistema, el sistema muestra la interface principal, luego hace click en “Registro de asistencia” y con un lector de barras realizará la lectura del código de barras del DNI el cual facilita el estudiante al momento de ingresar al local donde rendirá su examen, luego el sistema busca y registra la fecha y hora del DNI ingresado, seguidamente el sistema muestra en la pantalla el aula a cuál rendirá el examen y los datos del alumno como también su fotografía que registro el día de su matrícula. Si los datos son incorrectos el sistema muestra un mensaje de error.

Observación: Si el estudiante no pertenece al local donde ingreso el sistema muestra el local a donde pertenece y no registrar su asistencia.

Tabla 24.

T.I. – Control de aula en el examen

Tarea de ingeniería

Número de tarea de ingeniería: 6

Número de historia de usuario: 1

Nombre de la tarea: Gestionar asistencia en el aula donde rendirá el examen

Tipo de tarea: Desarrollo

Puntos estimados: 1

Fecha inicio: 13/06/2023

Fecha fin: 16/06/2023

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El operador accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña “registrar asistencia de aula”, el sistema muestra la interface “Registro de aula” y con un lector de código de barras realizará la lectura del DNI el cual facilita el responsable de aula al momento que el operador solicite aula por aula, luego el sistema busca y registra la fecha y hora del DNI ingresado, Seguidamente el sistema muestra en la pantalla con los datos del alumno. Si los datos son incorrectos el sistema muestra un mensaje de error.

Observación: El alumno no podrá registrar su registro de aula si no registro su ingreso al

local.

Tabla 25.

T. I. – Reporte de aulas designadas al operador

| Tarea de ingeniería | |
|--|---|
| Número de tarea de ingeniería: 7 | Número de historia de usuario: 1 |
| Nombre de la tarea: Generar reporte de registro de operadores | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 17/06/2023 | Fecha fin: 20/06/2023 |
| Programador responsable: Tueros Puchuri, Elmer | |
| Descripción: El operador accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña reportes, el sistema muestra la interface “reporte de aulas”, luego el sistema muestra las aulas disponibles que fueron asignadas al operador el día del examen, luego selecciona un aula, el sistema le muestra los alumnos pertenecientes al aula y los registros de los alumnos asistentes y faltantes, el sistema muestra si se registró el ingreso principal y el registro en el aula. | |
| Observación: ninguno | |

Tabla 26.

T.I. – Reporte de aulas designadas al auxiliar

| Tarea de ingeniería | |
|--|---|
| Número de tarea de ingeniería: 8 | Número de historia de usuario: 1 |
| Nombre de la tarea: Generar reporte de registro de auxiliar | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 21/06/2023 | Fecha fin: 23/06/2023 |

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El auxiliar accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña reportes, el sistema muestra la interface “reporte de aulas”, luego el sistema muestra las aulas disponibles que fueron asignadas al auxiliar para el ciclo académico, luego selecciona un aula, el sistema le muestra los alumnos pertenecientes al aula y los registros de los alumnos asistentes y faltantes, el sistema muestra si se registró de ingreso y el registro de salida.

Observación: ninguno

Tabla 27.

T.I. – Reporte general de la asistencia en clases

Tarea de ingeniería

Número de tarea de ingeniería: 9 y 10

Número de historia de usuario: 1

Nombre de la tarea: Generar reporte de las asistencias diarias

Tipo de tarea: Desarrollo

Puntos estimados: 1

Fecha inicio: 24/05/2023

Fecha fin: 27/05/2023

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El administrador accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña reportes, el sistema muestra la interface “reporte general”, luego el sistema muestra las aulas existentes, luego el administrador selecciona un aula, el sistema le muestra los alumnos pertenecientes al aula de clases y el registro de los alumnos asistentes y faltantes, el sistema muestra si se registró el ingreso principal y el registro en el aula.

Tabla 28.

T. I. – Reporte general de asistencia en el día del examen

Tarea de ingeniería

Número de tarea de ingeniería: 10

Número de historia de usuario: 1

Nombre de la tarea: Generar reporte de las asistencias al examen

Tipo de tarea: Desarrollo

Puntos estimados: 1

Fecha inicio: 28/06/2023

Fecha fin: 30/06/2023

Programador responsable: Tueros Puchuri, Elmer

Descripción:

El administrador accede al sistema, el sistema muestra la interface principal e ingresa a la pestaña reportes, el sistema muestra la interface “reporte general”, luego el sistema muestra las aulas existentes, luego el administrador selecciona un aula, el sistema le muestra los alumnos pertenecientes al aula de clases y el registro de los alumnos asistentes y faltantes, el sistema muestra si se registró el ingreso principal y el registro en el aula.

4.1.3.3. Plan de iteración

Establecemos los modelos de diversas iteraciones a lo largo de las versiones realizar para cada historia de usuario.

Tabla 29.

P.I. - Primera iteración

| N° | Historias de usuario | Tareas de ingeniería |
|-----------|---------------------------------|-----------------------------|
| 1 | Gestionar la cuenta del usuario | Crear cuenta de usuario |
| 2 | Gestionar sesión | Iniciar sesión |

Tabla 30.

P.I. - Segunda iteración

| N° | Historias de usuario | Tareas de ingeniería |
|-----------|---|-----------------------------|
| 1 | Gestionar asistencia diaria en la entrada del local de clases | Control de ingreso a clases |
| 2 | Gestionar asistencia diaria en la salida del local de clases | Control de salida de clases |

Tabla 31.*P.I. - Tercera iteración*

| N° | Historias de usuario | Tareas de ingeniería |
|-----------|---|------------------------------|
| 1 | Gestionar asistencia al ingresar al local el día del examen | Control de ingreso al examen |
| 2 | Gestionar asistencia en el aula donde rendirá el examen | Control de aula en el examen |

Tabla 32.*P.I. - Cuarta iteración*

| N° | Historias de usuario | Tareas de ingeniería |
|-----------|---|---|
| 1 | Generar reporte de registro de operadores | Reporte de aulas designadas al operador |
| 2 | Generar reporte de registro de auxiliar | Reporte de aulas designadas al auxiliar |

Tabla 33.*P.I. - Quinta iteración*

| N° | Historias de usuario | Tareas de ingeniería |
|-----------|--|--|
| 1 | Generar reporte de las asistencias diarias | Reporte general de la asistencia en clases |
| 2 | Generar reporte de las asistencias al examen | Reporte general de asistencia en el día del examen |

Plan de iteración clasificado en fechas de desarrollo**Tabla 34.***Plan de iteración clasificado en fechas*

| N° H.U. | Tareas de ingeniería | Fecha inicio | Fecha fin | Programador |
|----------------|-----------------------------|---------------------|------------------|-----------------------|
| 1 | 1 | 25/04/2022 | 27/04/2022 | Tueros Puchuri, Elmer |

| | | | | |
|----|---|------------|------------|-----------------------|
| 2 | 1 | 28/04/2022 | 30/04/2022 | Tueros Puchuri, Elmer |
| 3 | 1 | 01/06/2023 | 04/06/2023 | Tueros Puchuri, Elmer |
| 4 | 1 | 05/06/2023 | 08/06/2023 | Tueros Puchuri, Elmer |
| 5 | 1 | 09/06/2023 | 12/06/2023 | Tueros Puchuri, Elmer |
| 6 | 1 | 13/06/2023 | 16/06/2023 | Tueros Puchuri, Elmer |
| 7 | 1 | 17/06/2023 | 20/06/2023 | Tueros Puchuri, Elmer |
| 8 | 1 | 21/06/2023 | 23/06/2023 | Tueros Puchuri, Elmer |
| 9 | 1 | 24/06/2023 | 27/06/2023 | Tueros Puchuri, Elmer |
| 10 | 1 | 28/06/2023 | 30/06/2023 | Tueros Puchuri, Elmer |

4.1.3.4. Interfaz gráfica de usuario

Figura 4

Prototipo iniciar sesión

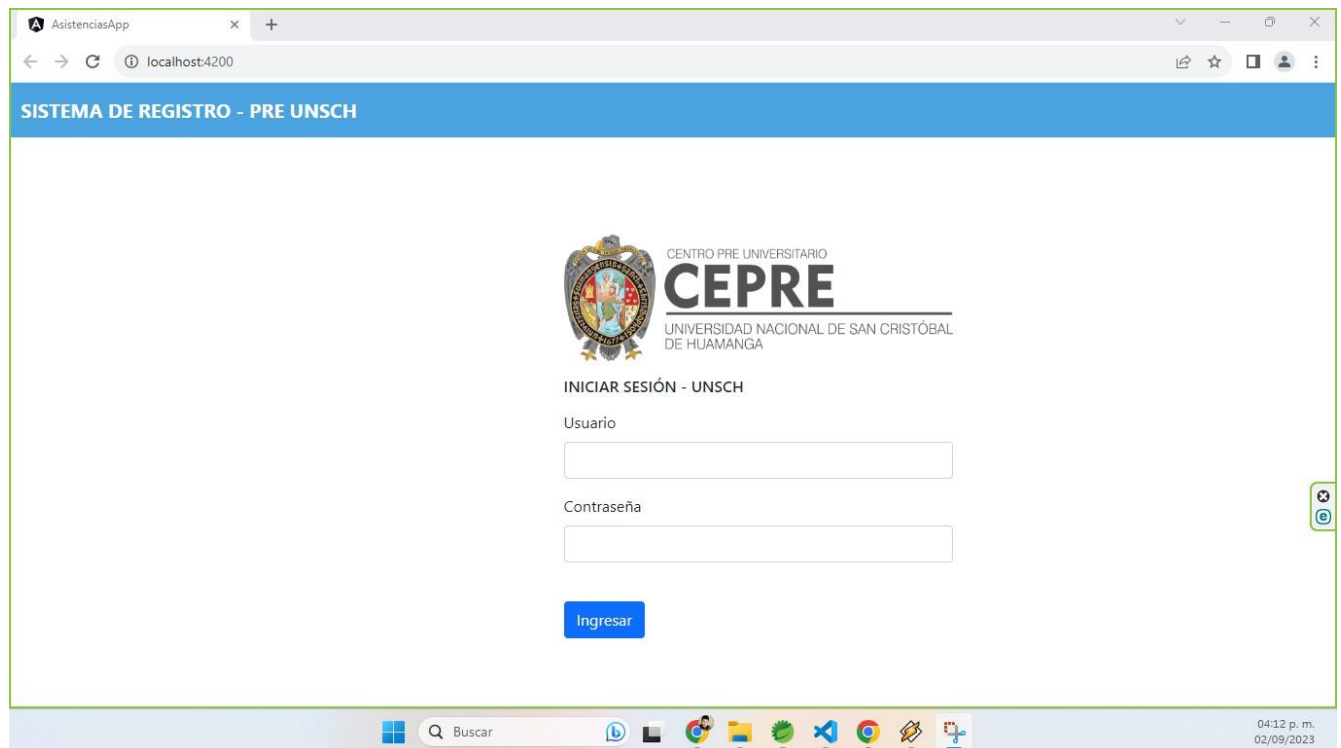


Figura 5

Pestaña de asistencia de clases

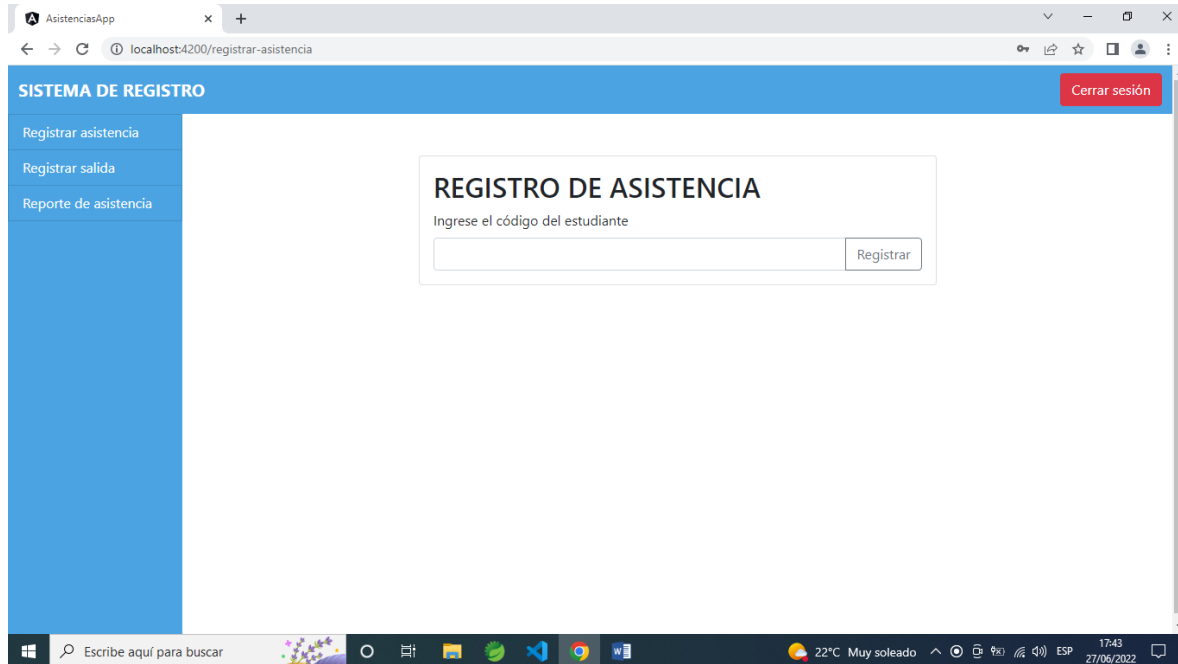


Figura 6

Registro exitoso de asistencia de clases

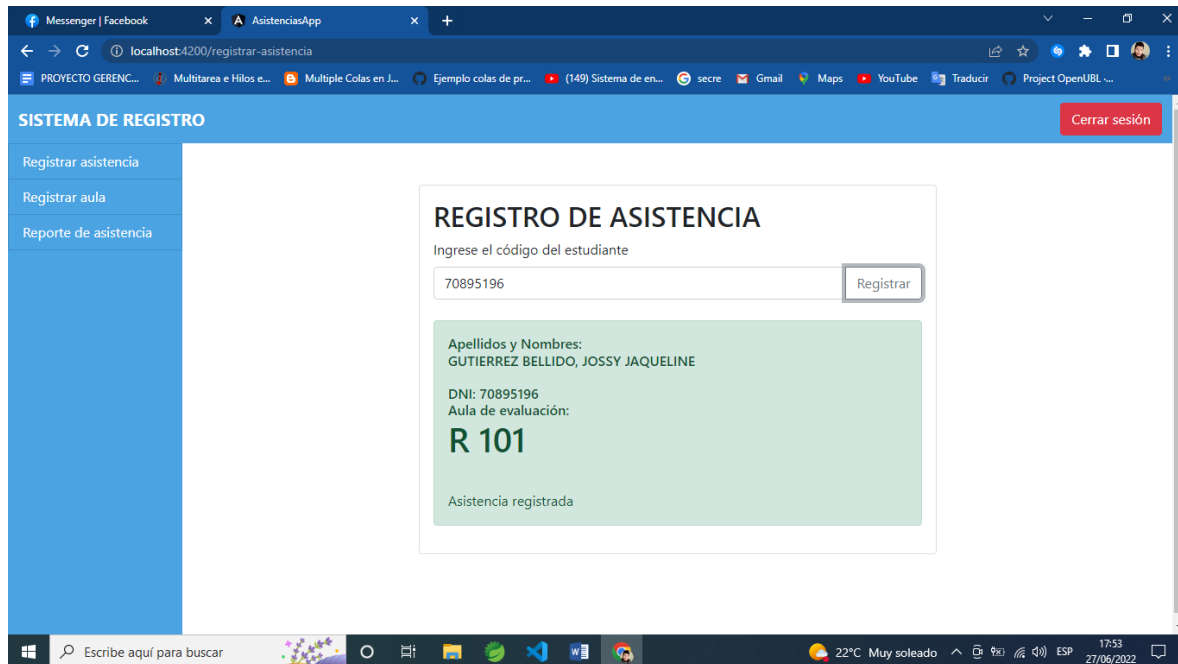


Figura 7

Alerta de registro en la asistencia a clases

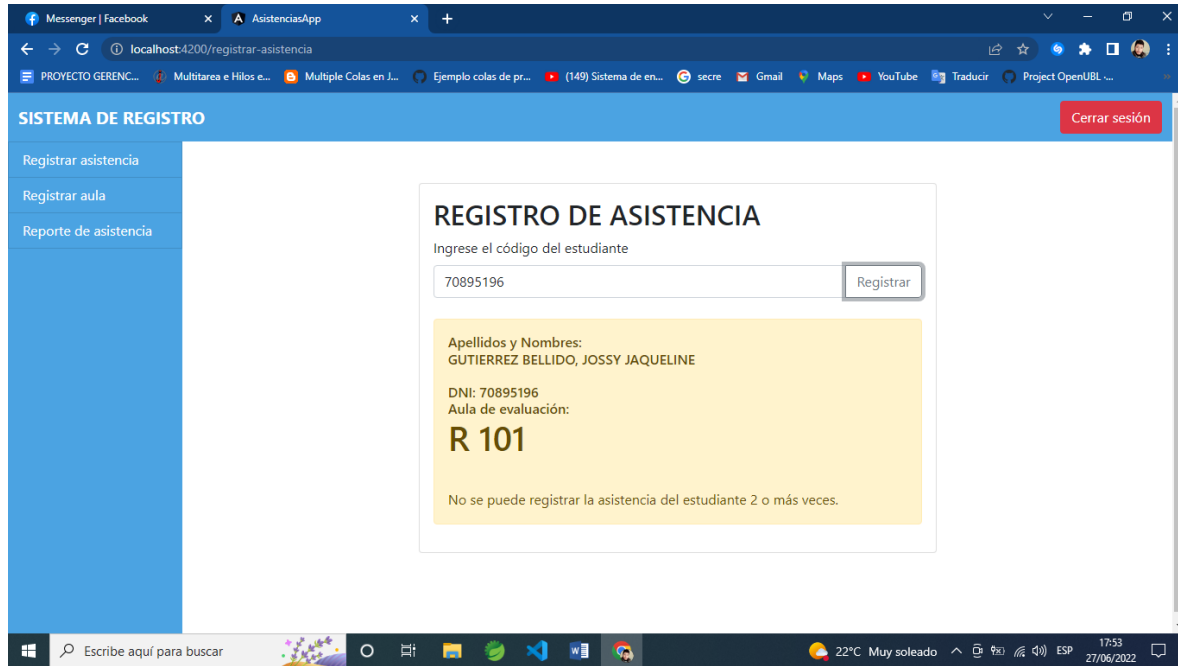


Figura 8

Pestaña de registro en aula para la salida de clases

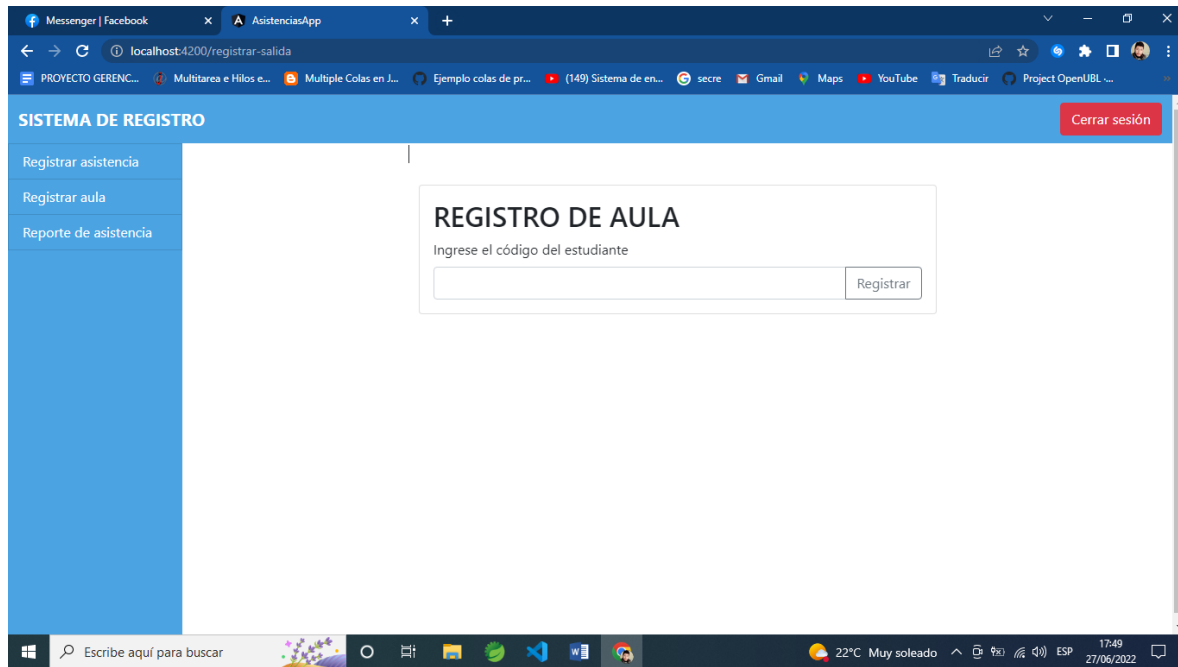


Figura 9

Registro exitoso en aula para la salida de clases

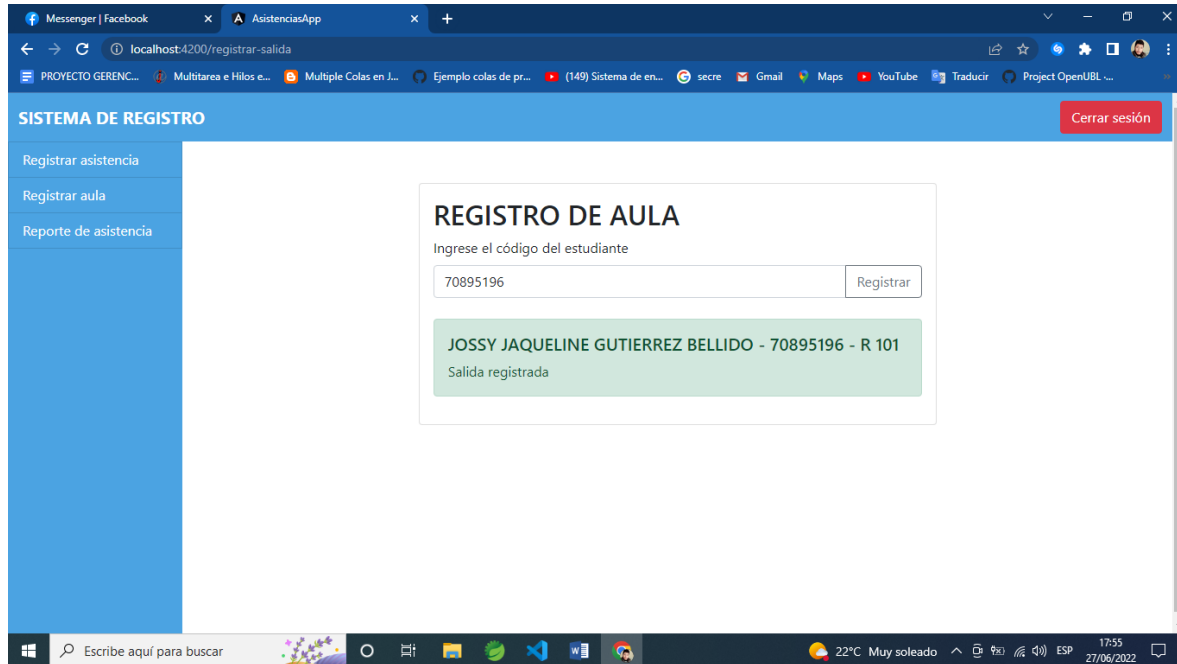


Figura 10

Alerta de registro en aula para la salida de clases

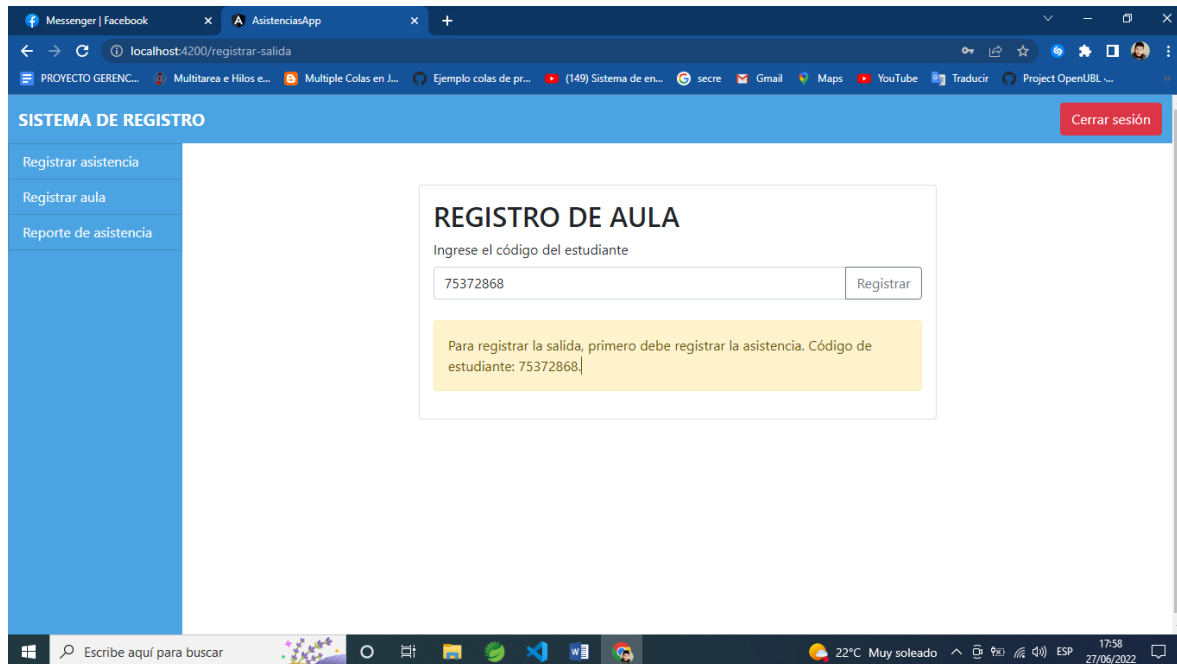


Figura 11

Reporte del auxiliar de asistencia a clases

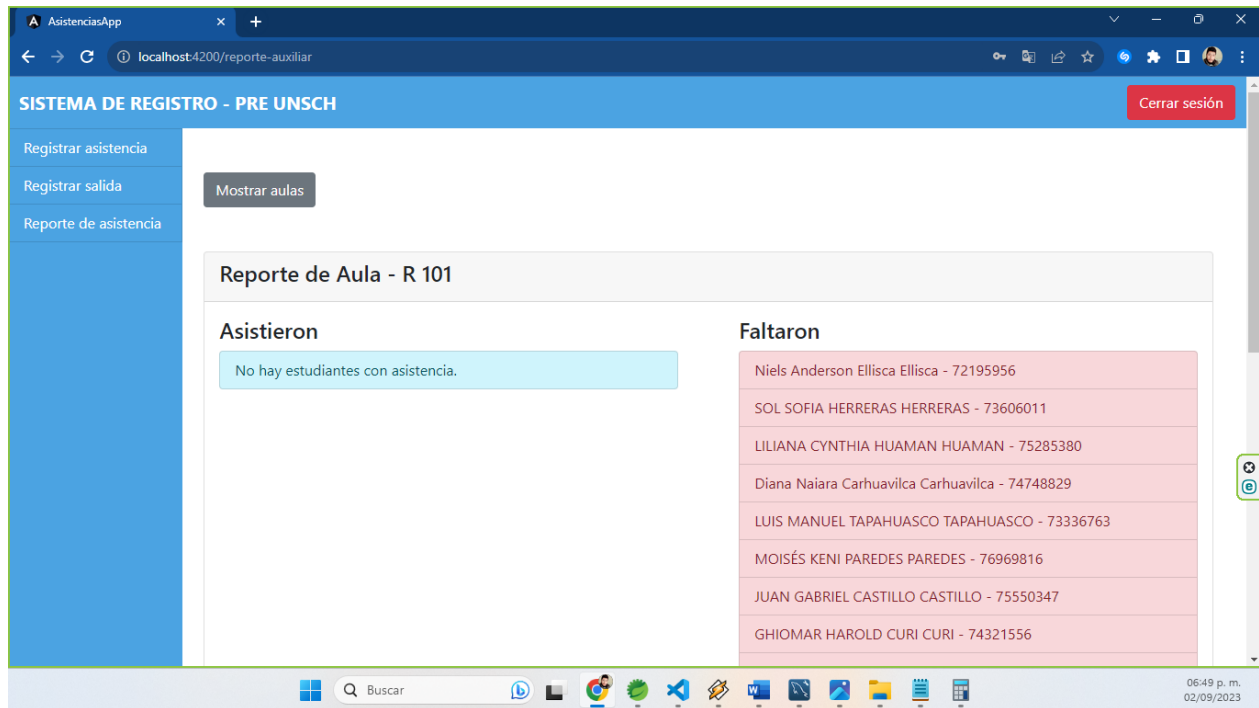


Figura 12

Asistencia a examen

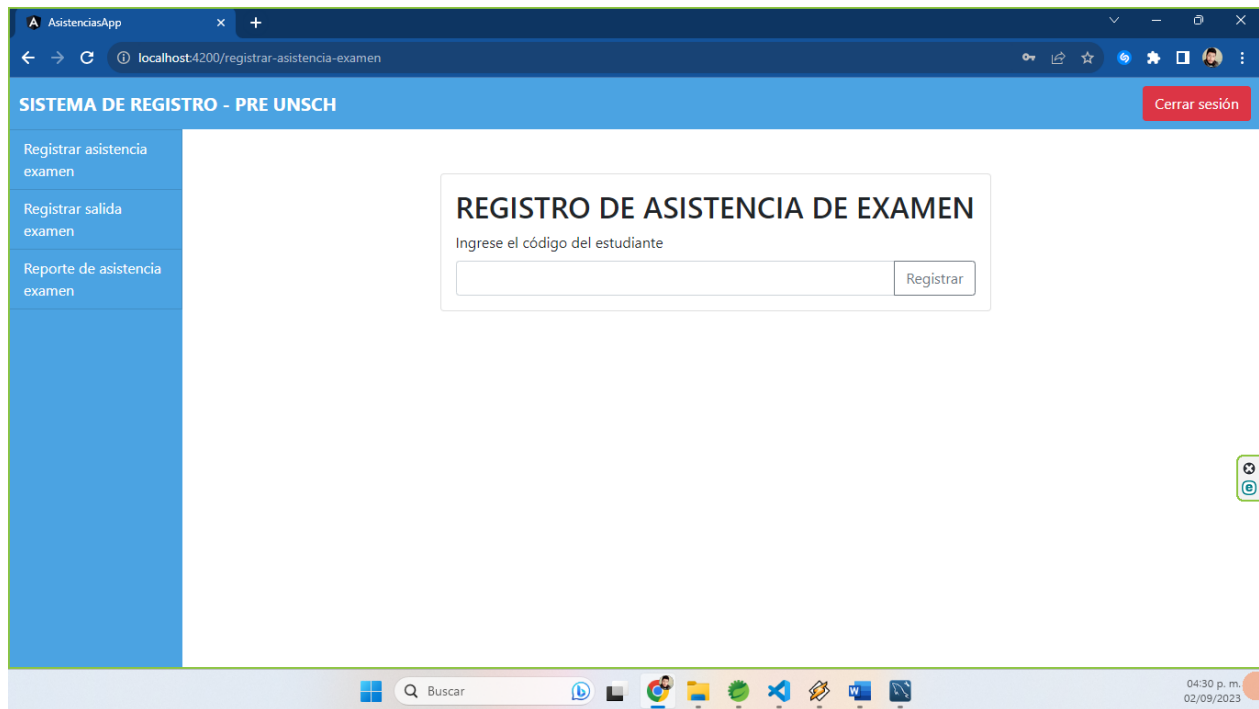


Figura 13

Registro exitoso de ingreso examen

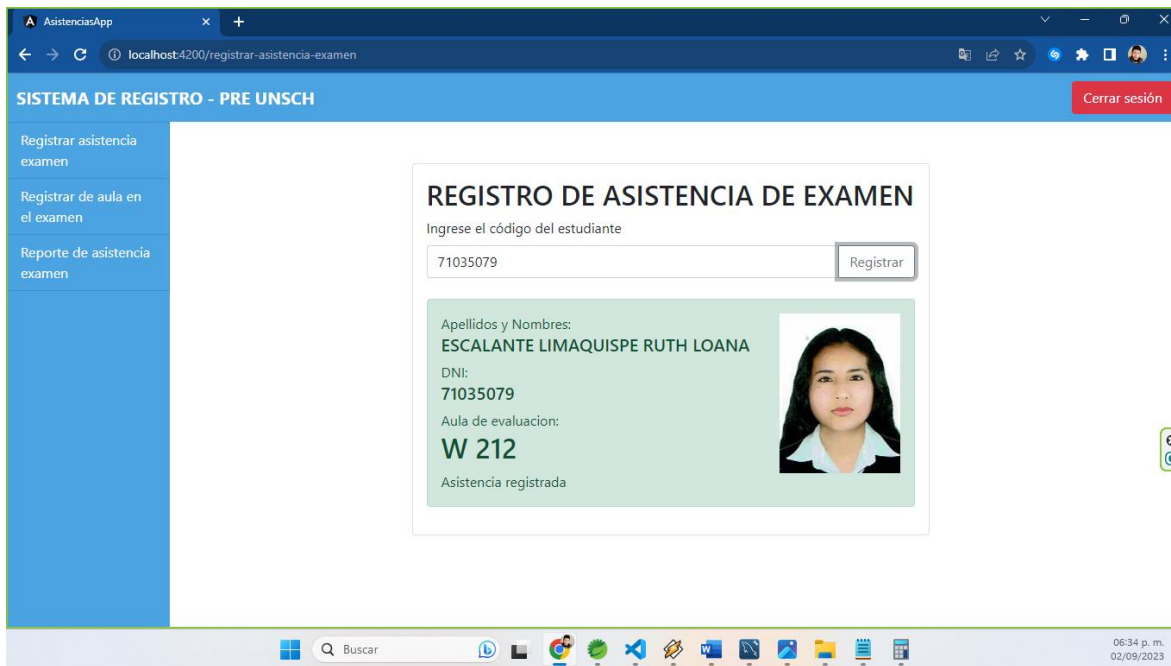


Figura 14

Alerta de registro en el ingreso al examen

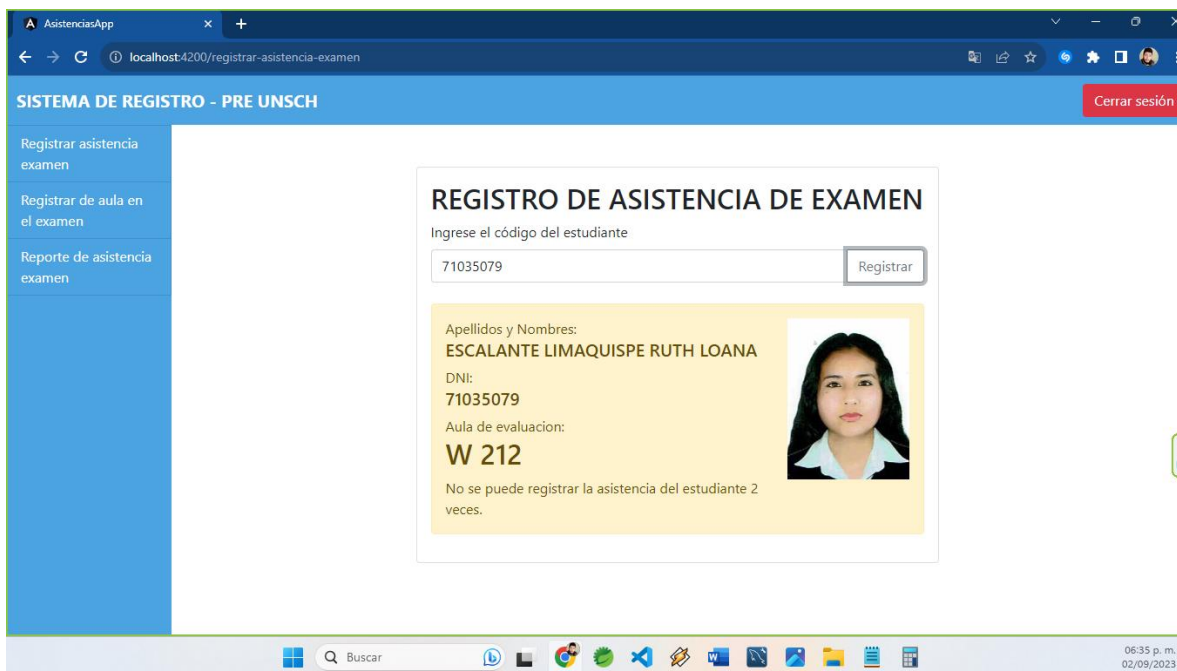


Figura 15

Pestaña de registro de asistencia al aula en el examen

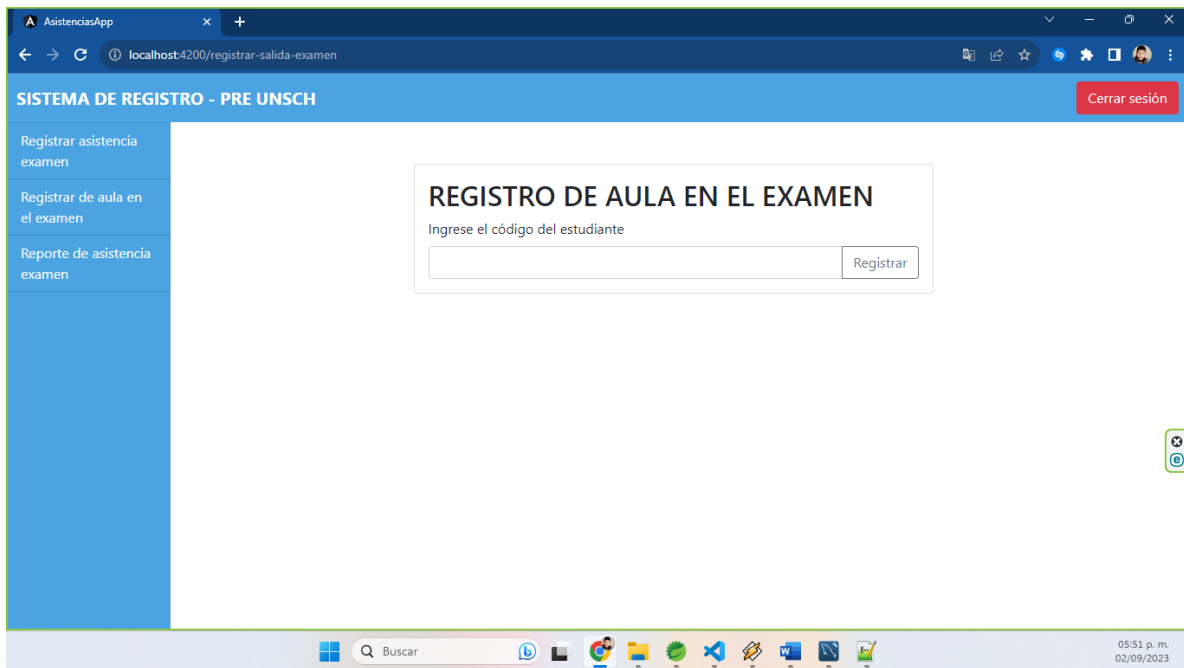


Figura 16

Registro exitoso en el aula del examen

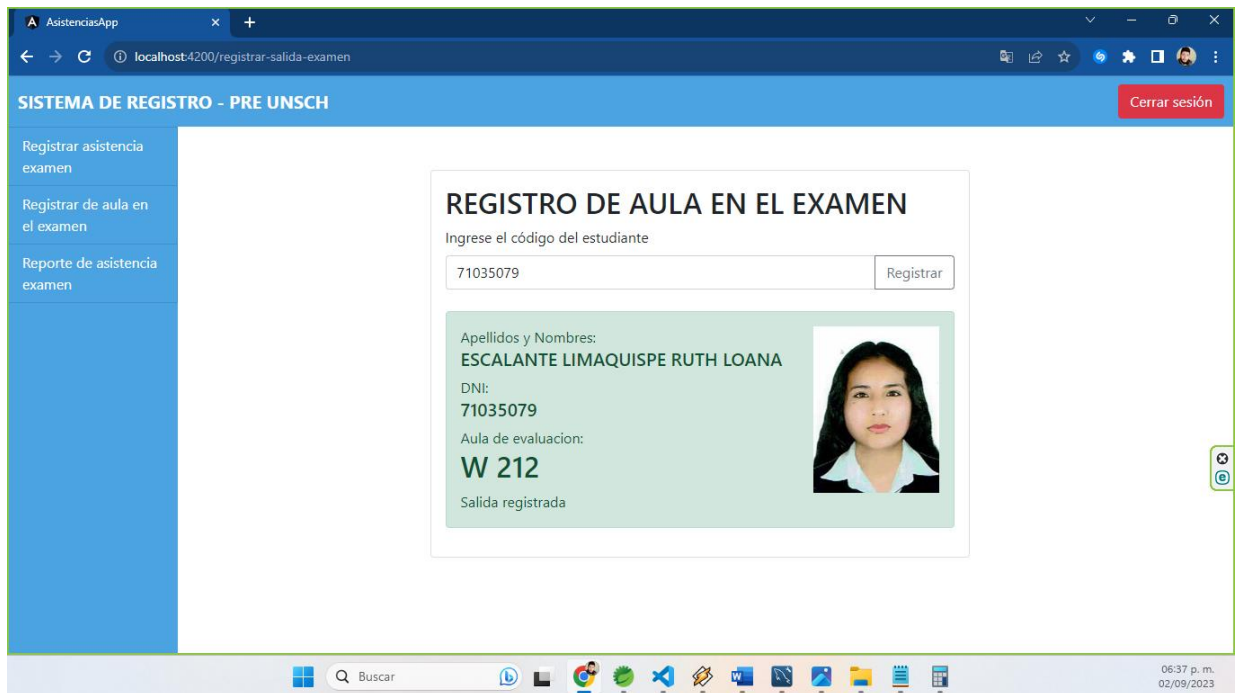


Figura 17

Alerta en el registro de asistencia en el aula del examen

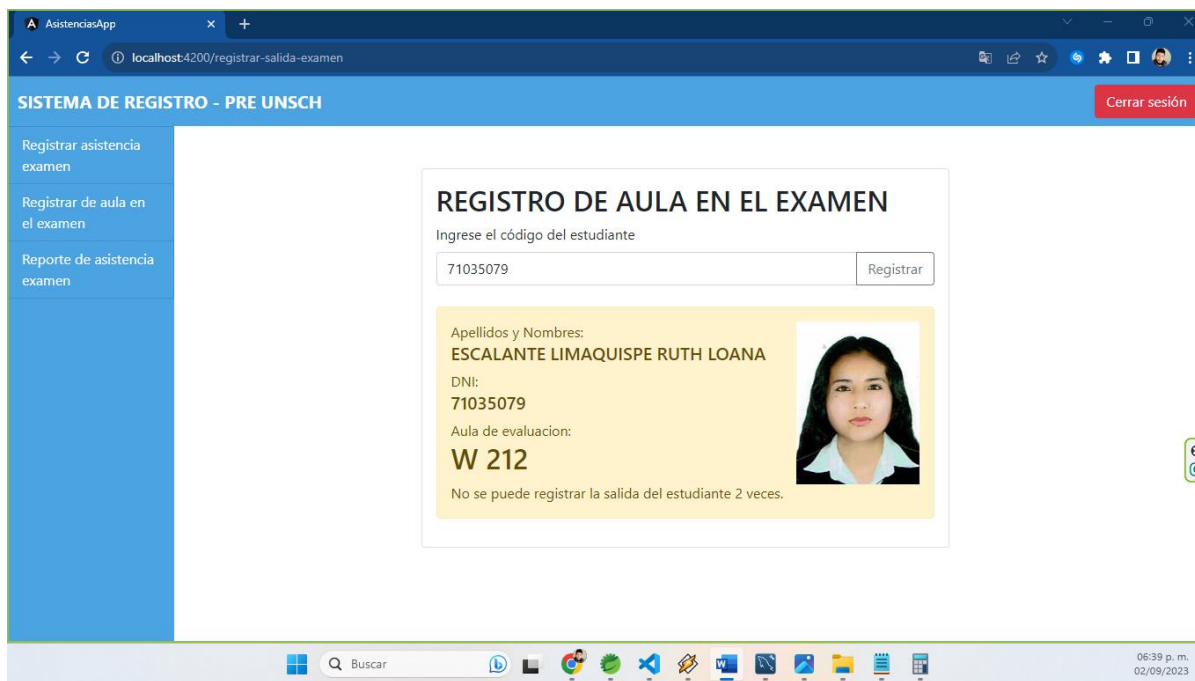


Figura 18

Reporte de asistentes al examen del operador

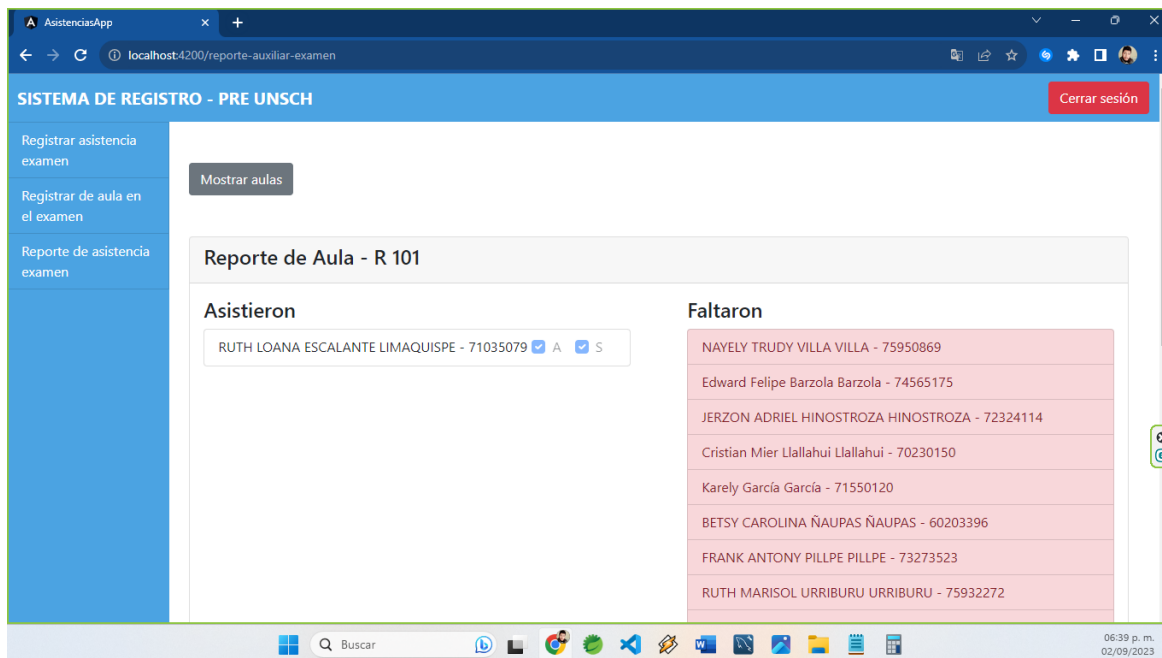


Figura 19

Reporte del administrador del aula de clases

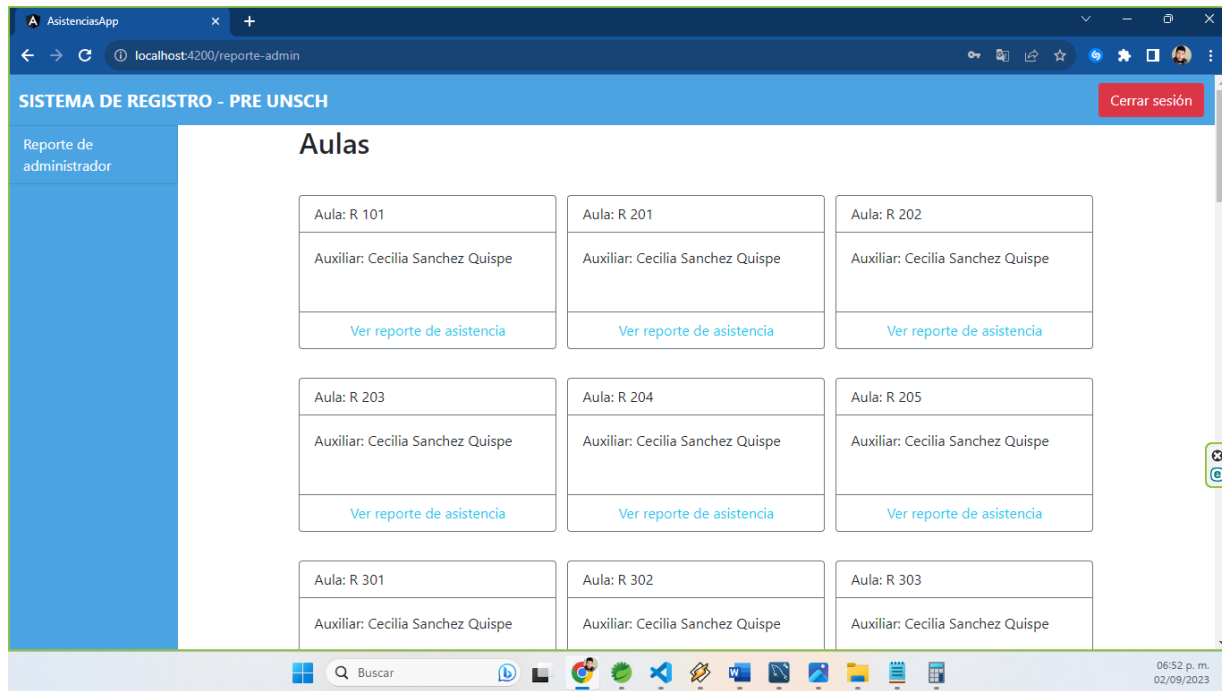
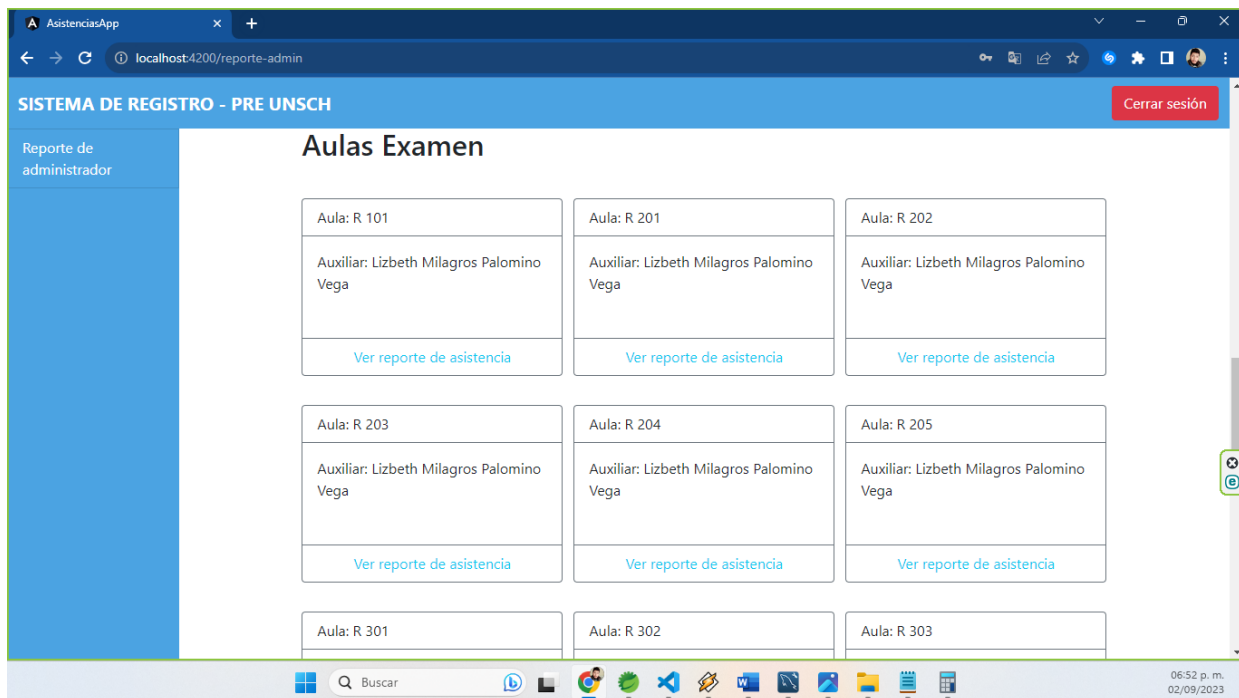


Figura 20

Reporte del administrador de las aulas del examen



4.1.3.5. Tarjeta CRC

Las tarjetas CRC permiten a los desarrolladores trabajar de forma orientada a objetos, contribuir al proceso de diseño y representar escenarios de historias de usuario nombrando clases de aplicaciones potenciales y asignando responsabilidades y colaboraciones. Estos escenarios luego se traducen en métodos y relaciones en iteraciones posteriores.

Tabla 35.

Clase Usuario

| Nombre de clase: usuario | |
|--|---|
| Responsabilidades: | Colaboradores: |
| <ul style="list-style-type: none">- Registrar usuario- Listar usuario | <ul style="list-style-type: none">- persona |

Tabla 36.

Clase Asistencia

| Nombre de clase: Asistencia | |
|--|--|
| Responsabilidades: | Colaboradores: |
| <ul style="list-style-type: none">- Buscar estudiante- Registrar estudiante- Listar estudiante | <ul style="list-style-type: none">- Usuarios registrados |

Tabla 37.

Clase Persona

| Nombre de clase: Persona | |
|--|--|
| Responsabilidades: | Colaboradores: |
| <ul style="list-style-type: none">- Buscar estudiante- Registrar estudiante- Asignar Rol | <ul style="list-style-type: none">- Operador- Auxiliar- Estudiante |

Tabla 38.

Clase Aula

| Nombre de clase: Aula | |
|--|--|
| Responsabilidades: | Colaboradores: |
| <ul style="list-style-type: none">- Buscar estudiante- Registrar estudiante- Asignar estudiantes | <ul style="list-style-type: none">- Usuarios registrados |

4.1.3.6. Código Fuente

Figura 21

Código de Spring Boot de la clase model - Aula

```
@AllArgsConstructor
@NoArgsConstructor
@Data
@Entity
public class Aula {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nombre;

    @ManyToOne
    @JsonIgnoreProperties({"aulas", "hibernateLazyInitializer",
"handler"})
    private Auxiliar auxiliar;

    @OneToMany(mappedBy = "aula")
    @JsonIgnoreProperties({"aula", "hibernateLazyInitializer", "handler"})
    private List<Estudiante> estudiantes;
}
```


Figura 22

Código de Spring Boot de la clase model - Asistencia

```

@Builder
@AllArgsConstructor
@NoArgsConstructor
@Data
@Entity
public class Asistencia {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @JsonFormat(pattern = "dd-MM-yyyy HH:mm:ss")
    private LocalDateTime fechaHoraIngreso;

    @JsonFormat(pattern = "dd-MM-yyyy HH:mm:ss")
    private LocalDateTime fechaHoraSalida;
    private Boolean ingresoConfirmado;
    private Boolean salidaConfirmado;
    private Boolean inasistencia;

    @JsonFormat(pattern = "dd-MM-yyyy")
    private LocalDate fechaQuery;

    @ManyToOne(fetch = FetchType.LAZY)
    private Estudiante estudiante;
}

```

Figura 23

Código de Spring Boot de la clase model - Estudiante

```

@AllArgsConstructor
@NoArgsConstructor
@Data
@Entity
public class Estudiante {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nombre;
    private String apellidoPaterno;
    private String apellidoMaterno;
    private String codigoEstudiante;
    private String imagen;

    @ManyToOne
    @JsonIgnoreProperties({"estudiantes", "hibernateLazyInitializer",
"handler"})
    @JoinColumn(name = "aula_id")
    private Aula aula;

    @ManyToOne
    @JsonIgnoreProperties({"estudiantes", "hibernateLazyInitializer",
"handler"})
    @JoinColumn(name = "aula_examen_id")
    private AulaExamen aulaExamen;
}

```

Figura 24

Código de Spring Boot de la clase service - AsistenciaServiceImpl

```

@RequiredArgsConstructor
@Service
public class AsistenciaServiceImpl implements AsistenciaService{
    private final AsistenciaRepo asistenciaRepo;
    private final AulaRepo aulaRepo;

    @Override
    public Asistencia guardar(Asistencia asistencia) throws
ServiceException {

```

```

        if(!asistencia.getIngresoConfirmado() ||
asistencia.getSalidaConfirmado()){
            throw new ServiceException("Parámetros inválidos");
        }try {
            LocalDateTime fechaActual = LocalDateTime.now();
            LocalDate fechaQuery = LocalDate.now();
            asistencia.setFechaHoraIngreso(fechaActual);
            asistencia.setFechaQuery(fechaQuery);
            return asistenciaRepo.save(asistencia);
        } catch (DataAccessException e) {
            throw new ServiceException("Error en la base de datos al
insertar la asistencia.");
        }
    }
    @Override
    public Optional<Asistencia> buscarPorId(Long id) {
        return asistenciaRepo.findById(id);
    }
    @Override
    public List<Asistencia> buscarTodo() {
        return asistenciaRepo.findAll();
    }
    @Override
    public Optional<Asistencia> buscarPorEstudianteCodigo(String
codigoEstudiante) {
        LocalDate fechaActual = LocalDate.now();
        return
asistenciaRepo.findByIdByEstudiante_CodigoEstudianteAndFechaQuery(codigoEstudi
ante, fechaActual);
    }
    @Override
    public Asistencia registrarSalida(Asistencia asistencia) {
        LocalDateTime fechaActual = LocalDateTime.now();
        asistencia.setFechaHoraSalida(fechaActual);
        return asistenciaRepo.save(asistencia);
    }
    @Override
    public List<Asistencia> obtenerReporteAuxiliar(boolean ingreso) {
        LocalDate fechaActual = LocalDate.now();
        return
asistenciaRepo.findByIdByFechaQueryAndIngresoConfirmadoAndEstudiante_Aula_Id(fe
chaActual, ingreso,1L);
    }
}

```

Figura 25

Código de Spring Boot de la clase service - *AulaExamenServiceImpl*

```

@RequiredArgsConstructor
@Service
public class AulaExamenServiceImpl implements AulaExamenService {
    private final AulaExamenRepo aulaRepo;
    @Override
    public AulaExamen guardar(AulaExamen aulaExamen) throws
ServiceException {
        try {
            return aulaRepo.save(aulaExamen);
        } catch (DataAccessException e) {
            throw new ServiceException("Error en la base de datos al
insertar el aula.");
        }
    }
    @Override
    public Optional<AulaExamen> buscarPorId(Long id) {
        return aulaRepo.findById(id);
    }
    @Override
    public List<AulaExamen> buscarTodo() {
        return aulaRepo.findAll();
    }
    @Override
    public Optional<Estudiante> buscarEstudiantePorCodigo(String
codigoEstudiante) {
        return aulaRepo.buscarEstudiantePorCodigo(codigoEstudiante);
    }
    @Override
    public List<AulaExamen> buscarAulasExamenPorAuxiliar(Long idAuxiliar)
{
        return aulaRepo.buscarAulasExamenPorAuxiliar(idAuxiliar);
    }
}

```

Figura 26

Código de Spring Boot de la clase service - AuxiliarServiceImpl

```

@RequiredArgsConstructor
@Service
public class AuxiliarServiceImpl implements AuxiliarService {
    private final AuxiliarRepo auxiliarRepo;

    @Override
    public Auxiliar guardar(Auxiliar auxiliar) throws ServiceException {
        try {
            return auxiliarRepo.save(auxiliar);
        } catch (DataAccessException e) {
            throw new ServiceException("Error en la base de datos al
insertar el auxiliar.");
        }
    }
    @Override
    public Optional<Auxiliar> buscarPorId(Long id) {
        return auxiliarRepo.findById(id);
    }
    @Override
    public List<Auxiliar> buscarTodo() {
        return auxiliarRepo.findAll();
    }
    @Override
    public Optional<Auxiliar> buscarPorUsuarioAndPassword(String usuario,
String password) {
        return auxiliarRepo.findByUsuarioAndPassword(usuario, password);
    }
}

```

Figura 27

Código de Spring Boot de la clase controller - AulaController

```

@RestController
@CrossOrigin
@RequestMapping("/api/aulas")
public class AulaController {

```

```

private AulaService aulaService;
public AulaController(AulaService aulaService) {
    this.aulaService = aulaService;
}
@GetMapping("/obtener-aulas")
public ResponseEntity<?> obtenerAulas() {
    return ResponseEntity
        .status(OK)
        .body(CustomResponse.builder()
            .code(SUCCESS)
            .message("Transacción exitosa!")
            .data(aulaService.buscarTodo()).build());
}
@GetMapping("/buscar/{id}")
public ResponseEntity<?> buscarAula(@PathVariable Long id) {
    Optional<Aula> aula = aulaService.buscarPorId(id);

    if(!aula.isPresent()) {
        return ResponseEntity
            .status(NOT_FOUND)
            .body(CustomResponse.builder()
                .message("No se encontró un aula con ese ID: "+id)
                .code(WARNING).build());
    }
    return ResponseEntity
        .status(OK)
        .body(CustomResponse.builder()
            .code(SUCCESS)
            .message("Transacción exitosa!")
            .data(aulaService.buscarPorId(id)).build());
}
@GetMapping("/por-auxiliar/{idAuxiliar}")
public ResponseEntity<?> buscarAulasPorAuxiliar(@PathVariable Long
idAuxiliar) {
    List<Aula> aulas =
aulaService.buscarAulasExamenPorAuxiliar(idAuxiliar);
    return ResponseEntity
        .status(OK)
        .body(CustomResponse.builder()
            .code(SUCCESS)
            .message("Transacción exitosa!")
            .data(aulas).build());
}
@PostMapping("/registrar")
public ResponseEntity<?> registrarAula(@RequestBody Aula aula) throws
ServiceException {
    return new ResponseEntity<>(CustomResponse.builder()
        .code(SUCCESS).message("Aula registrada")
        .data(aulaService.guardar(aula)).build(), CREATED);
}
}

```

Figura 28

Código de Spring Boot de la clase controller - AuxiliarController

```
@RestController
@CrossOrigin
@RequestMapping("/api/auxiliares")
public class AuxiliarController {
    private AuxiliarService auxiliarService;

    public AuxiliarController(AuxiliarService auxiliarService) {
        this.auxiliarService = auxiliarService;
    }
    @PostMapping("/registrar")
    public ResponseEntity<?> registrarAuxiliar(@RequestBody Auxiliar
auxiliar) throws ServiceException {
        return new ResponseEntity<>(CustomResponse.builder()
            .code(SUCCESS).message("Auxiliar registrado")
            .data(auxiliarService.guardar(auxiliar)).build(),
CREATED);
    }
    @GetMapping("/{id}")
    public ResponseEntity<?> registrarAuxiliar(@PathVariable Long id) {
        Auxiliar auxiliar = auxiliarService.buscarPorId(id).get();
        return new ResponseEntity<>(CustomResponse.builder()
            .code(SUCCESS).message("Auxiliar registrado")
            .data(auxiliar).build(), OK);
    }
    @GetMapping("/login")
    public ResponseEntity<?> ingresarAlSistema(@PathParam("usuario")
String usuario, @PathParam("password") String password) {
        Optional<Auxiliar> auxiliar =
auxiliarService.buscarPorUsuarioAndPassword(usuario, password);
        if(!auxiliar.isPresent()) {
            return ResponseEntity
                .status(NOT_FOUND)
                .body(CustomResponse.builder()
                    .message("Usuario y/o contraseña incorrectos.")
                    .code(WARNING).build());
        }
        return ResponseEntity
            .status(OK)
            .body(CustomResponse.builder()
                .message("Login exitoso")
                .code(SUCCESS).data(auxiliar.get()).build());
    }
}
```

Figura 29

Código Angular – auxiliar.service.ts

```
● ● ●  
  
@Injectable({  
  providedIn: 'root'  
})  
export class AuxiliarService {  
  
  private urlBackend: string = environment.urlBackend+'/auxiliares';  
  constructor(private http: HttpClient, private router: Router) { }  
  
  login(usuario: string, password: string): Observable<any> {  
    let params = new HttpParams();  
    params = params.set('usuario', usuario);  
    params = params.set('password', password);  
    return this.http.get<any>(`${this.urlBackend}/login`, {params:  
params});  
  }  
  
  buscarPorId(id: number | null | string): Observable<any> {  
    return this.http.get<any>(`${this.urlBackend}/${id}`);  
  }  
  cerrarSesion() {  
    window.sessionStorage.clear();  
    this.router.navigate(['login'])  
  }  
}
```

Figura 30

Código Angular – asistencia.service.ts

```
● ● ●  
  
@Injectable({  
  providedIn: 'root'  
})  
export class AsistenciaService {
```



```

private urlBackend: string = environment.urlBackend+'/asistencias';
private _sendLoginStatusSource = new Subject<any>();
loginStatus$ = this._sendLoginStatusSource.asObservable();
constructor(private http: HttpClient) { }
registrarAsistencia(asistencia: Asistencia): Observable<any> {
    return this.http.post<any>(`${this.urlBackend}/registrar-asistencia`,
asistencia);
}

registrarSalida(asistencia: Asistencia): Observable<any> {
    return this.http.put<any>(`${this.urlBackend}/registrar-salida`,
asistencia);
}

reporteAsistencia(clase_id: string): Observable<any> {
    return this.http.get<any>(`${this.urlBackend}/reporte`);
}

buscarAsistenciaPorCodigoEstudiante(codigoEstudiante: string):
Observable<any> {
    return this.http.get<any>(`${this.urlBackend}/buscar-por-
codigo/${codigoEstudiante}`);
}

buscarAsistenciaExamenPorCodigoEstudiante(codigoEstudiante: string):
Observable<any> {
    return this.http.get<any>(`${this.urlBackend}-examen/buscar-por-
codigo/${codigoEstudiante}`);
}

public setUsuario(usuario: any): void {
    window.sessionStorage.removeItem('USUARIO');
    window.sessionStorage.setItem('USUARIO', usuario);
}

public getUsuario(): any {
    return sessionStorage.getItem('USUARIO');
}

sendLoginStatus( usuario: any | undefined) {
    this._sendLoginStatusSource.next(usuario);
}
}

```

Figura 31

Código Angular componente – Registrar asistencia

```
Component({
  selector: 'app-registrar-asistencia',
  templateUrl: './registrar-asistencia.component.html',
  styleUrls: ['./registrar-asistencia.component.css']
})
export class RegistrarAsistenciaComponent implements OnInit {
  asistencia!: Asistencia | undefined;
  estudiante: Estudiante = new Estudiante();
  mensaje!: string | undefined;
  mostrarDatosAsistencia: boolean = false;
  codigoRespuesta!: string | undefined;
  constructor(private asistenciaService: AsistenciaService) { }
  ngOnInit(): void {
  }
  registrarAsistencia() {
    this.asistencia = new Asistencia();
    this.asistencia.estudiante = this.estudiante;
    this.asistencia.ingresoConfirmado=true;
    this.asistencia.salidaConfirmado=false;
    this.mensaje = undefined;
    this.codigoRespuesta = undefined;
    this.mostrarDatosAsistencia = false;
    this.asistenciaService.registrarAsistencia(this.asistencia)
      .subscribe({
        next: (response) => {
          this.asistencia = response.data;
          console.log(response.code);
          this.mensaje = response.message;
          this.mostrarDatosAsistencia = true;
          this.codigoRespuesta = response.code;
        },
        error: (err) => {
          this.mostrarDatosAsistencia = true;
          this.asistencia = err.error.data;
          this.mensaje = err.error.message;
          console.log(err)
        }
      })
  }
}
```

Figura 32

Código Angular componente – Registrar de salida a clases

```
Component({
  selector: 'app-registrar-salida',
  templateUrl: './registrar-salida.component.html',
  styleUrls: ['./registrar-salida.component.css']
})
export class RegistrarSalidaComponent implements OnInit {
  asistencia!: Asistencia | undefined;
  estudiante: Estudiante = new Estudiante();
  mensaje!: string | undefined;
  mostrarDatosAsistencia: boolean = false;
  codigoRespuesta!: string | undefined;
  constructor(private asistenciaService: AsistenciaService) { }
  ngOnInit(): void {
  }
  registrarSalida() {
    this.asistencia = new Asistencia();
    this.asistencia.estudiante = this.estudiante;
    this.asistencia.ingresoConfirmado=true;
    this.asistencia.salidaConfirmado=true;
    this.mensaje = undefined;
    this.codigoRespuesta = undefined;
    this.mostrarDatosAsistencia = false;
    this.asistenciaService.registrarSalida(this.asistencia)
    .subscribe({
      next: (response) => {
        this.asistencia = response.data;
        console.log(response.code);
        this.mensaje = response.message;
        this.mostrarDatosAsistencia = true;
        this.codigoRespuesta = response.code;
      },
      error: (err) => {
        this.mostrarDatosAsistencia = true;
        this.asistencia = err.error.data;
        this.mensaje = err.error.message;
      }
    })
  }
}
```

4.1.3.7. Pruebas unitarias

Aquí se describen las pruebas unitarias utilizadas para garantizar que las tareas técnicas se completen de forma correcta y confiable.

Tabla 39.

P.I. - Primera iteración

| N° | Historias de usuario | Tareas de ingeniería | Resultado |
|-----------|---|--|------------------|
| 1 | Gestionar la cuenta del usuario | Crear cuenta de usuario | Exitoso |
| 2 | Gestionar sesión | Iniciar sesión | Exitoso |
| 3 | Gestionar asistencia diaria en la entrada del local de clases | Control de ingreso a clases | Exitoso |
| 4 | Gestionar asistencia diaria en la salida del local de clases | Control de salida de clases | Exitoso |
| 5 | Gestionar asistencia al ingresar al local el día del examen | Control de ingreso al examen | Exitoso |
| 6 | Gestionar asistencia en el aula donde rendirá el examen | Control de aula en el examen | Exitoso |
| 7 | Generar reporte de registro de operadores | Reporte de aulas designadas al operador | Exitoso |
| 8 | Generar reporte de registro de auxiliar | Reporte de aulas designadas al auxiliar | Exitoso |
| 9 | Generar reporte de las asistencias diarias | Reporte general de la asistencia en clases | Exitoso |
| 10 | Generar reporte de las asistencias al examen | Reporte general de asistencia en el día del examen | Exitoso |

4.1.3.8. Pruebas de aceptación

Aquí detallamos las diversas pruebas de aceptación realizadas para garantizar que se satisfagan todas las necesidades de nuestro cliente.

Tabla 40.

P. A. Crear cuenta de usuario

| Prueba de aceptación | |
|--|---|
| N° PA: 1 | Historia de Usuario: Gestionar la cuenta del usuario |
| Nombre de la prueba: Crear usuario | |
| Descripción: El sistema solicitara el usuario y la contraseña al administrador, este debe introducir estos campos para poder dar acceso al sistema a nuevos usuarios. | |
| Condiciones de ejecución: | |
| <ul style="list-style-type: none"> - El administrador es el encargado de otorgar cuentas a los demás usuarios | |
| Entrada: | |
| <ul style="list-style-type: none"> - El administrador debe ingresar, los datos del usuario, DNI, clave y el rol, luego click en registrar | |
| Resultado esperado: | |
| <ul style="list-style-type: none"> - El aplicativo guarda la información en la Base de Datos - Solo el usuario registrado tiene permiso de acceso al sistema | |
| Evaluación de prueba: Prueba satisfactoria | |

Tabla 41.

P. A. Iniciar sesión

| Prueba de aceptación | |
|--|--|
| N° PA: 2 | Historia de Usuario: Gestionar sesión |
| Nombre de la prueba: Autenticación | |
| Descripción: El sistema debe solicitar y verificar el usuario para el acceso al sistema. El usuario debe introducir usuario y contraseña para poder tener acceso al sistema. | |
| Condiciones de ejecución: | |
| <ul style="list-style-type: none"> - El usuario debe estar registrado en la Base de datos | |
| Entrada: | |
| <ul style="list-style-type: none"> - El usuario inicia el sistema introduciendo su usuario y contraseña y luego hace clic en ingresar | |
| Resultado esperado: | |
| <ul style="list-style-type: none"> - El sistema realiza la consulta en la base de datos y si existiese permite el acceso | |

-
- Solo el usuario registrado podrá acceder al sistema
-

Evaluación de prueba: Prueba satisfactoria

Tabla 42.

P. A. Control de ingreso a clases

| Prueba de aceptación | |
|---|---|
| N° PA: 3 | Historia de Usuario: Gestionar asistencia diaria en la entrada del local de clases |
| Nombre de la prueba: Registro de asistencia a clases | |
| Descripción: La aplicación debe realizar la búsqueda y el registro de la fecha y hora de ingreso del alumno | |
| Condiciones de ejecución: Debe estar el auxiliar autenticado para poder registrar | |
| Entrada: <ul style="list-style-type: none">- El auxiliar ingresa en el menú de registrar asistencia.- El auxiliar registra con un lector de barras y el DNI del estudiante. | |
| Resultado esperado: <ul style="list-style-type: none">- El sistema muestra los datos del alumno con el registro satisfactorio. | |
| Evaluación de prueba: Prueba satisfactoria | |

Tabla 43.

P.A. Control de salida de clases

| Prueba de aceptación | |
|---|--|
| N° PA: 4 | Historia de Usuario: Gestionar asistencia diaria en la salida del local de clases |
| Nombre de la prueba: Registro de salida de clases | |
| Descripción: La aplicación debe realizar la búsqueda y el registro de la fecha y hora de salida de clases del alumno | |
| Condiciones de ejecución: Debe estar el auxiliar autenticado para poder registrar | |
| Entrada: <ul style="list-style-type: none">- El auxiliar ingresa en el menú de registrar salida.- El auxiliar registra con un lector de barras y el DNI del estudiante. | |

Resultado esperado:

- El sistema muestra los datos del alumno con el registro satisfactorio.

Evaluación de prueba: Prueba satisfactoria

Tabla 44.

P. A. Control de ingreso al examen

| Prueba de aceptación | |
|---|---|
| N° PA: 5 | Historia de Usuario: Gestionar asistencia al ingresar al local el día del examen |
| Nombre de la prueba: Registro de ingreso al examen | |
| Descripción: La aplicación debe realizar la búsqueda y el registro de la fecha y hora de ingreso del alumno | |
| Condiciones de ejecución: Debe estar el operador autenticado para poder registrar | |
| Entrada: <ul style="list-style-type: none">- El operador ingresa en el menú de registrar asistencia.- El operador registra con un lector de barras y el DNI del estudiante. | |
| Resultado esperado: <ul style="list-style-type: none">- El sistema muestra los datos del alumno con el registro satisfactorio. | |
| Evaluación de prueba: Prueba satisfactoria | |

Tabla 45.

P.A. Control de aula en el examen

| Prueba de aceptación | |
|--|---|
| N° PA: 6 | Historia de Usuario: Gestionar asistencia en el aula donde rendirá el examen |
| Nombre de la prueba: Registro del alumno en aula | |
| Descripción: La aplicación debe realizar la búsqueda y el registro de la fecha y hora del estudiante presneten en el aula del examen | |
| Condiciones de ejecución: Debe estar el operador autenticado para poder registrar | |
| Entrada: <ul style="list-style-type: none">- El operador ingresa en el menú de registrar salida. | |

-
- El operador registra con un lector de barras y el DNI del estudiante.
-

Resultado esperado:

- El sistema muestra los datos del alumno con el registro satisfactorio.
-

Evaluación de prueba: Prueba satisfactoria

Tabla 46.

P.A. Reporte de aulas designadas al operador

| Prueba de aceptación | |
|---|---|
| N° PA: 7 | Historia de Usuario: Generar reporte de registro de operadores |
| Nombre de la prueba: Reporte de asistencia al examen | |
| Descripción: La aplicación permitirá mostrar el reporte de los alumnos registrados en el ingreso como también en el aula | |
| Condiciones de ejecución: Debe estar el operador autenticado para poder monitorear | |
| Entrada: <ul style="list-style-type: none">- El operador ingresa en el menú de reportes y debe seleccionar el aula a visualizar. | |
| Resultado esperado: <ul style="list-style-type: none">- El sistema muestra los datos del alumno en el aula correspondiente. | |
| Evaluación de prueba: Prueba satisfactoria | |

Tabla 47.

P.A. Reporte de aulas designadas al auxiliar

| Prueba de aceptación | |
|---|---|
| N° PA: 8 | Historia de Usuario: Generar reporte de registro de auxiliar |
| Nombre de la prueba: Reporte de asistencia de las clases presenciales | |
| Descripción: La aplicación permitirá mostrar el reporte de los alumnos registrados del alumno en su respectiva aula | |
| Condiciones de ejecución: Debe estar el auxiliar autenticado para poder monitorear | |
| Entrada: | |

-
- El auxiliar ingresa en el menú de reportes y debe seleccionar el aula a visualizar.
-

Resultado esperado:

- El sistema muestra los datos del alumno en el aula correspondiente.
-

Evaluación de prueba: Prueba satisfactoria

Tabla 48.

P.A. Reporte general de la asistencia en clases

| Prueba de aceptación | |
|--|--|
| N° PA: 9 | Historia de Usuario: Generar reporte de las asistencias diarias |
| Nombre de la prueba: Reporte de asistencia al examen | |
| Descripción: La aplicación permitirá mostrar el reporte de todas las aulas de los alumnos registrados al ingreso como también de la salida del local de clases | |
| Condiciones de ejecución: Debe estar el administrador autenticado para poder monitorear | |
| Entrada: <ul style="list-style-type: none">- El administrador ingresa en el menú de reportes y debe seleccionar el aula a visualizar. | |
| Resultado esperado: <ul style="list-style-type: none">- El sistema muestra todas las aulas de todos los locales correspondiente al ciclo académico. | |
| Evaluación de prueba: Prueba satisfactoria | |

Fuente: Elaboración propia

Tabla 49.

P.A. Reporte general de asistencia en el día del examen

| Prueba de aceptación | |
|--|--|
| N° PA: 10 | Historia de Usuario: Generar reporte de las asistencias al examen |
| Nombre de la prueba: Reporte de asistencia al examen | |
| Descripción: La aplicación permitirá mostrar el reporte de todas las aulas de los alumnos registrados al ingreso como también en el aula del local del examen. | |

Condiciones de ejecución: Debe estar el administrador autenticado para poder monitorear

Entrada:

- El administrador ingresa en el menú de reportes y debe seleccionar el aula a visualizar.
-

Resultado esperado:

- El sistema muestra todas las aulas de todos los locales correspondiente al ciclo académico.
-

Evaluación de prueba: Prueba satisfactoria

4.1.3.9. Resumen de pruebas de aceptación

Tabla 50.

Resumen de pruebas de aceptación

| N° | Historias de usuario | N° C. P. | Caso de prueba (C.P.) |
|------------------|---|-----------------|---|
| H.U. 1 | Gestionar la cuenta del usuario | 1 | La aplicación permite el correcto registro de las cuentas de usuarios y permisos necesarios que estarán definidos por los siguientes roles; administrador, auxiliar y operador. |
| 2 | Gestionar sesión | 2 | La aplicación verifica correctamente la autenticación de usuario y contraseña que le ha sido designado al usuario. |
| 3 | Gestionar asistencia diaria en la entrada del local de clases | 3 | La aplicación permite a los auxiliares el correcto registro del ingreso de los estudiantes al momento de su ingreso a los locales donde se les impartirá clases presenciales |
| 4 | Gestionar asistencia diaria en la salida del local de clases | 4 | La aplicación permite a los auxiliares el correcto registro del estudiante cuando estos se retiren de los locales donde se les imparten las clases presenciales |
| 5 | Gestionar asistencia al ingresar al local el día del examen | 5 | La aplicación permite al operador el correcto registro del ingreso del estudiante al momento de ingresar al local del examen. |

| | | | |
|----|---|----|--|
| 6 | Gestionar asistencia en el aula donde rendirá el examen | 6 | La aplicación permite al operador el correcto registro de la asistencia del estudiante en sus respectivas aulas designadas para rendir el examen. |
| 7 | Generar reporte de registro de operadores | 7 | La aplicación permite al operador visualizar adecuadamente el reporte del registro de entrada como el registro en aula de los estudiantes designados para su local en el día del examen. |
| 8 | Generar reporte de registro de auxiliar | 8 | La aplicación permite al auxiliar visualizar adecuadamente el reporte del registro de entrada como el registro en salida de los estudiantes designados para su local en y sus respectivas aulas. |
| 9 | Generar reporte de las asistencias diarias | 9 | La aplicación permite al administrador visualizar adecuadamente la asistencia de todas las aulas asignadas a los estudiantes que asisten a clases. |
| 10 | Generar reporte de las asistencias al examen | 10 | La aplicación permite al administrador visualizar adecuadamente la asistencia de todas las aulas asignadas a los estudiantes que asisten el día del examen. |

Tabla 51.

Reporte de pruebas de aceptación

| N° H.U. | N° C.P. | Caso de prueba (C.P.) | Resultado |
|------------|------------|---|-----------|
| 1 | 1 | La aplicación permite el correcto registro de las cuentas de usuarios y permisos necesarios que estarán definidos por los siguientes roles; administrador, auxiliar y operador. | Exitoso |
| 2 | 2 | La aplicación verifica correctamente la autenticación | Exitoso |

| | | | |
|----|----|--|---------|
| | | de usuario y contraseña que le ha sido designado al usuario. | |
| 3 | 3 | La aplicación permite a los auxiliares el correcto registro del ingreso de los estudiantes al momento de su ingreso a los locales donde se les impartirá clases presenciales | Exitoso |
| 4 | 4 | La aplicación permite a los auxiliares el correcto registro del estudiante cuando estos se retiren de los locales donde se les imparten las clases presenciales | Exitoso |
| 5 | 5 | La aplicación permite al operador el correcto registro del ingreso del estudiante al momento de ingresar al local del examen. | Exitoso |
| 6 | 6 | La aplicación permite al operador el correcto registro de la asistencia del estudiante en sus respectivas aulas designadas para rendir el examen. | Exitoso |
| 7 | 7 | La aplicación permite al operador visualizar adecuadamente el reporte del registro de entrada como el registro en aula de los estudiantes designados para su local en el día del examen. | Exitoso |
| 8 | 8 | La aplicación permite al auxiliar visualizar adecuadamente el reporte del registro de entrada como el registro en salida de los estudiantes designados para su local en y sus respectivas aulas. | Exitoso |
| 9 | 9 | La aplicación permite al administrador visualizar adecuadamente la asistencia de todas las aulas asignadas a los estudiantes que asisten a clases. | Exitoso |
| 10 | 10 | La aplicación permite al administrador visualizar adecuadamente la asistencia de todas las aulas asignadas a los estudiantes que asisten el día del examen. | Exitoso |

CAPÍTULO V

CONCLUSIONES

5.1. CONCLUSIONES

- a. Se automatizo la asistencia en aula, se presenta los artefactos más importantes mediante las historias de usuario de las tablas 10, 11 y 15; como también las tareas de ingeniería en las tablas 21, 22 y 27; interfaces graficas mediante las figuras 5, 8 y 11, como el diseño de la base de datos física mediante el anexo 3.
- b. Se automatizo la asistencia en aula, se presenta los artefactos más importantes mediante las historias de usuario de las tablas 12, 13 y 14; como también las tareas de ingeniería en las tablas 23, 24 y 28; interfaces graficas mediante las figuras 14, 15 y 18, como el diseño de la base de datos física mediante el anexo 3.

CAPÍTULO VI

RECOMENDACIONES

6.1. RECOMENDACIONES

- a. Se recomienda crear una API RestFul, para historias de usuario académicas, que agilice la asistencia y la gestión de datos de los alumnos, para evitar la redundancia de datos.
- b. Se recomienda agregar otras historias de usuario al software, como la asistencia del personal, para el monitoreo durante el proceso de exámenes.
- c. Desarrollar una aplicación web integrada a las aplicaciones de la Universidad Nacional de San Cristóbal de Huamanga, para obtener información en tiempo real.

Referencias bibliográficas

- Beck, K. (2000). *Extreme Programming Explained. Embrace Change*. (W. Addison, Trans.) Pearson Education. (Trabajo original publicado en 1999)
- Canós, J. H., Letelier, P., & Penadés, M. C. (2003). *Metodologías ágiles en el desarrollo de software*. Universidad Politécnica de Valencia, 1-8. <http://www.aleteya.cs.buap.mx/~jlavalle/papers/agileMethodology/TodoAgil.pdf>
- Calabria, L., & Piriz, P. (2003). *Metodología XP*. Universidad ORT Uruguay, Facultad de Ingeniería.
- Capó Marrero, L. y García Dorta, D. (2013). Sistema de Gestión de solicitudes de descargas para la red de la Universidad Agraria de La Habana. Researchgate https://www.researchgate.net/publication/295902780_Sistema_de_Gestion_de_solicitudes_de_descargas_para_la_red_de_la_Universidad_Agraria_de_La_Habana
- Dipp Sejas, C. (2019). *Características de angular y react para desarrollo en interfaces de usuario de aplicaciones web*. [Tesis Doctoral, Universidad Mayor de San Simón]. <http://ddigital.umss.edu.bo:8080/jspui/handle/123456789/14760>
- Grupo Edebé. (21 de diciembre de 2009). *Gestión integral de recursos humanos*. Edebe. https://www.edebe.com/ciclosformativos/zona-publica/ut03_830029_la_oarh_cas.pdf
- Haimann Theo, S. W. (2005). *En Dirección y gerencia: planificación, coordinación y control de las actividades de la empresa*. Hispano Europea Barcelona
- Hurtado Pardo, E. F., & Llanos Bermudez, J. E. (2021). *Sistema de control de asistencia a estudiantes mediante carnet Virtual con código QR*. [Tesis Pregrado, Universidad distrital Francisco José de Caldas]. <https://repository.udistrital.edu.co/handle/11349/26731>
- Joskowicz, J. (2008). *Reglas y prácticas en Extreme Programming*. Vigo.
- Kendall, E. y Kendall, J. (2005). *Análisis y diseño de sistemas*. (6° Ed.). México: Pearson Education.
- Koontz, H., Weihrich, H., & Cannice, M. (2012). *Sistema y Proceso de Control. En Administración una perspectiva Global y Empresarial*. McGraw Hill.
- Letelier, P. y Sánchez, E. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Alicante, España: Grupo ISSI. <https://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
- Lomas Fernández, J. M. (2022). *Sistema de Control de Asistencia (SAM)*. [Tesis Pregrado Universidad de Cantabria]. <https://repositorio.unican.es/xmlui/handle/10902/27446>
- Luján, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Madrid, España: Club Universitario.

- Medinilla, A., Giné, A., y Gómez, E. (2001). *Manifiesto por el Desarrollo Ágil de Software: Agilmanifesto*. Recuperado el 04 setiembre de 2019 de <http://agilemanifesto.org/iso/es/manifiesto.html>
- Meliá Beigbeder, S. (2007). *Un Método de Desarrollo Dirigido por Modelos de Arquitectura para Aplicaciones Web*. [Tesis Doctoral, Universidad de Alicante] <https://dialnet.unirioja.es/servlet/dctes?codigo=68508>
- Sommerville, I. (2022). *Ingeniería del software*. Pearson educación.
- Sulla Hinojosa, T. (2022). *Sistema biométrico basado en aplicaciones móviles para el control de asistencia de estudiantes del Instituto Superior Tecnológico Americana del Cusco*. [Tesis Pregrado, Universidad Cesar Vallejo] <https://repositorio.ucv.edu.pe/handle/20.500.12692/87748>
- Pérez Ramírez, D. (2015). Metodologías ágiles: ¿Cómo desarrollo utilizando XP?. *Convención científica de ingeniería y arquitectura*, 1-5.
- Porras Flores, E. E. (2010). *Comparación de dos procesos de desarrollo de software usando los métodos ICONIX y XP, caso: comercialización de la Tara de la Región Ayacucho*. [Tesis Maestría, Universidad Nacional de Ingeniería] <http://hdl.handle.net/20.500.14076/14802>
- Real Academia Española. (s.f.). Absentismo. *En Diccionario de la lengua española*. Recuperado en 20 de marzo de 2023, de <https://www.rae.es/drae2001/absentismo>
- Rodríguez Echevarría, R., Sosa Sánchez, E. & Prieto Ramos, A. (2004). *Programación orientada a objetos*. <http://190.57.147.202:90/xmlui/handle/123456789/1419>
- Rodríguez, H. y Vergara, J. (2004). *Arquitectura de Software. Model, View and Controller*. Los protocolos en las redes de ordenadores.
- Rosa Moncayo, J. M. (17 de mayo de 2018) *¿Que es REST conoce su potencia?*. Openwebinars. <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>
- Seoane, E. (2005). *La nueva era del comercio: el comercio electrónico*. Editorial S.L.
- Viera Palacios, Z. Y. E. (2020). *Implementación de un sistema de información para reducir errores en el proceso de control de asistencias del personal de las instituciones educativas que pertenecen a la UGEL-Huamanga de la región Ayacucho*. [Tesis Pregrado, Universidad de Piura] <https://repositorio.unp.edu.pe/handle/20.500.12676/2419>
- Zulian, E. R. (2011). *Implementación de un framework para el desarrollo de aplicaciones web utilizando patrones de diseño y arquitectura MVC/REST* [Tesis Doctoral, Universidad de Belgrano] <http://190.221.29.250/handle/123456789/640>

Anexos

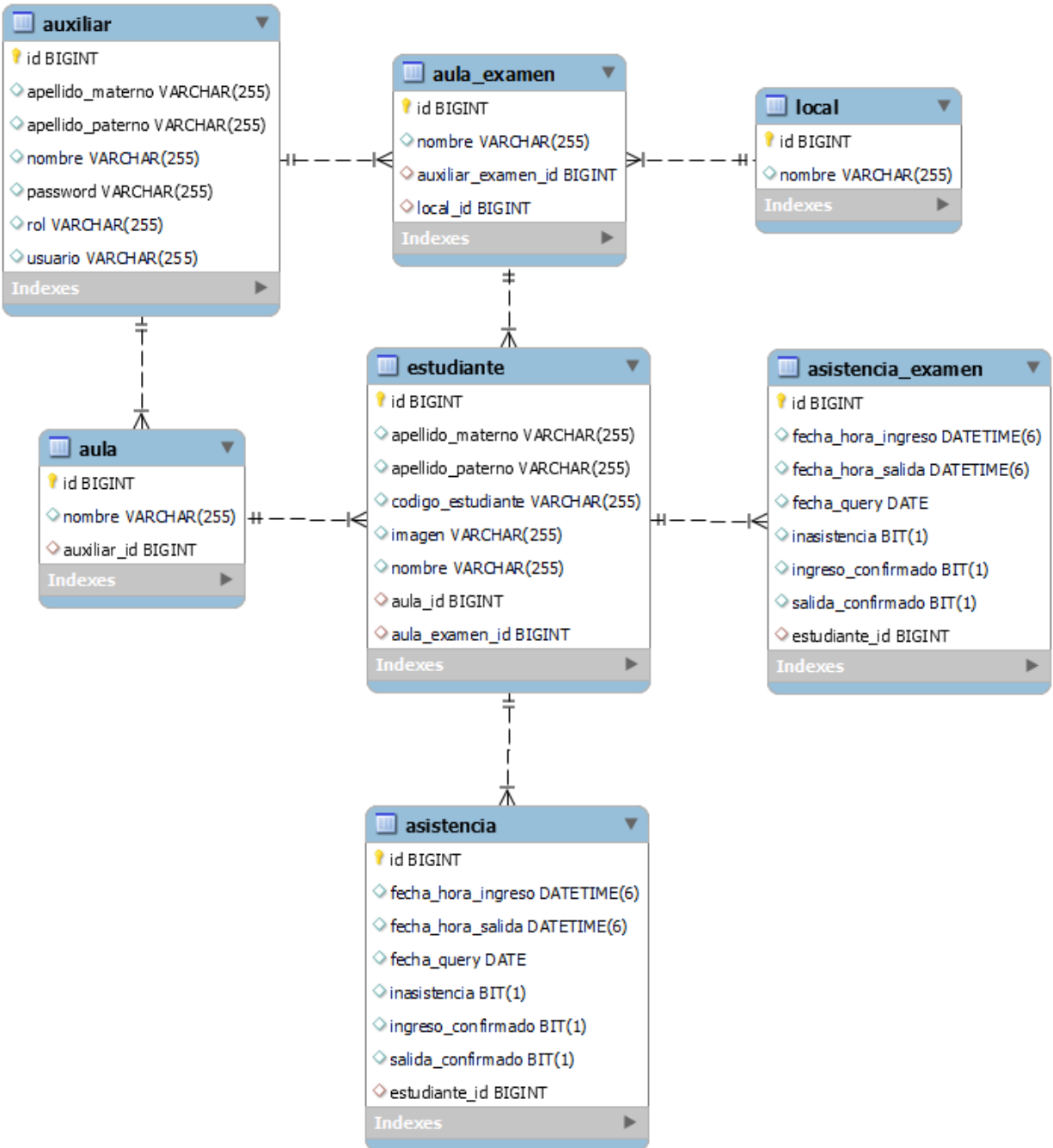
Anexo 1. Matriz de consistencia

Título: Aplicación Web para Control de Asistencia de Estudiantes en el Centro Preuniversitario de la UNSCH, Región Ayacucho, 2023

| PROBLEMAS | OBJETIVOS | VARIABLES | METODOLOGÍA |
|---|--|---|--|
| <p>PROBLEMA GENERAL ¿Cómo automatizar el control de asistencia de los Estudiantes en el Centro Preuniversitario de la UNSCH, Región Ayacucho, 2023?</p> <p>PROBLEMAS ESPECÍFICOS a. ¿De qué manera automatizar el control de asistencia en aula? b. ¿De qué manera automatizar el control de asistencia a exámenes?</p> | <p>OBJETIVO GENERAL Desarrollar una aplicación web para el control de asistencia de los Estudiantes en el Centro Preuniversitario de la UNSCH, Región Ayacucho, 2023</p> <p>OBJETIVOS ESPECÍFICOS a. Explorar, planificar e iterar para automatizar el control de asistencia en aula b. Explorar, planificar e iterar para automatizar el control de asistencia a exámenes</p> | <p>VARIABLE DE INTERÉS X: Control de asistencia</p> <p>VARIABLES DESCRIPTIVAS X1: Asistencia en aula X2: Asistencia en examen</p> | <p>TIPO DE INVESTIGACIÓN Observacional, prospectivo, transversal y descriptivo</p> <p>NIVEL DE INVESTIGACIÓN Descriptivo</p> <p>DISEÑO No experimental, prospectivo-transversal</p> <p>POBLACIÓN La población está compuesta por 3 procesos de control de asistencia en aula y 3 de control de asistencia a exámenes, de estudiantes</p> |

| | | | |
|--|--|--|--|
| | | | <p>del Centro preuniversitario de la Universidad Nacional de San Cristóbal e Huamanga el año 2023</p> <p>MUESTRA No existe muestra, será un censo, porque se estudiará 3 procesos de asistencia en aula y 3 procesos de asistencia en examen.</p> <p>TÉCNICA Entrevista</p> <p>INSTRUMENTO Guía de entrevista</p> |
|--|--|--|--|

Anexo 2: Modelo de Base de Datos Física



Anexo 3: Guía de Entrevista

**“APLICACIÓN WEB PARA CONTROL DE ASISTENCIA DE ESTUDIANTES EN EL CENTRO
PREUNIVERSITARIO DE LA UNSCH, REGIÓN AYACUCHO, 2023”**

Fecha: ____/____/____

Nombre del entrevistado: _____

Institución: _____

Objetivo:

Desarrollar una aplicación web para el control de asistencia de los Estudiantes en el Centro Preuniversitario de la UNSCH, Región Ayacucho.

Preguntas:

1. ¿Cómo el estudiante se registra al momento del ingreso a las instalaciones donde se impartirán las clases?

2. ¿Cuántas inasistencias debe tener el estudiante para llamarle la atención?

3. ¿Cómo toman decisiones a consecuencia de un alto porcentaje de inasistencia?

4. ¿Cómo verifica la inasistencia en tiempo real?

5. ¿Qué tipo de control general de los estudiantes existe?

6. ¿Cómo se monitorea la puntualidad de los estudiantes?

7. ¿Qué herramienta digital se necesita para facilitar el registro de ingreso?

8. ¿Cómo registrar la asistencia de estudiantes en los exámenes?

9. ¿Qué herramienta facilita el registro de asistencia en aula para el examen?

10. ¿De qué manera al ingresar a los locales del examen se toma la asistencia a los estudiantes?

11. ¿Cuándo ingresan al aula, donde se rendirá el examen, se toma asistencia?

12. ¿Cómo verifica la inasistencia en tiempo real a los exámenes?

13. ¿Cómo verifica la identidad del estudiante que rendirá el examen?

14. ¿Cómo verifica que el estudiante que se encuentra en su aula, el día del examen paso por los filtros de ingreso?

15. ¿Qué herramienta digital permite monitorear y facilitar el registro de los estudiantes el día del examen?

**ACTA DE SUSTENTACIÓN DE TESIS****ACTA N° 075-2023-FIMGC**

En la ciudad de Ayacucho, en cumplimiento a la **RESOLUCIÓN DECANAL N° 353-2023-FIMGC-D**, siendo ocho días del mes de setiembre del 2023, a horas 8:00 am.; se reunieron los jurados del acto de sustentación, en el Auditorium virtual google meet del Campus Universitario de la Universidad Nacional de San Cristóbal de Huamanga.

Siendo el Jurado de la sustentación de tesis compuesto por el presidente el **Dr. Ing. Manuel Avelino LAGOS BARZOLA**, Jurado el **Mg. Ing. Eloy VILA HUAMAN**, Jurado el **Mg. Ing. Javier PORTILLO QUISPE**, Jurado - Asesor el **Dr. Ing. Efraín Elías PORRAS FLORES** y secretario del proceso el **Mg. Ing. Christian LEZAMA CUELLAR**, con el objetivo de recepcionar la sustentación de la tesis denominada titulado: **“APLICACIÓN WEB PARA CONTROL DE ASISTENCIA DE ESTUDIANTES EN EL CENTRO PREUNIVERSITARIO DE LA UNSCH, REGIÓN AYACUCHO, 2023”**, presentado por el/la Sr./Srta., **ELMER TUEROS PUCHURI**, Bachiller en **Ingeniería de sistemas**.

El Jurado luego de haber recepcionado la sustentación de la tesis y realizado las preguntas, el sustentante al haber dado respuesta a las preguntas, y el Jurado haber deliberado; califica con la nota aprobatoria de **15 (quince)**.

En fe de lo cual, se firma la presente acta, por los miembros integrantes del proceso de sustentación.

Dr. Ing. Manuel Avelino LAGOS BARZOLA
Presidente

Mg. Ing. Eloy VILA HUAMAN
Jurado

Dr. Ing. Efraín Elías PORRAS FLORES
Jurado Asesor

Firmado digitalmente por Dr.
Ing. Efraín Elías Porras Flores
Fecha: 2023.09.08 19:14:53
-05'00'

Mg. Ing. Javier PORTILLO QUISPE
Jurado

Mg. Ing. Christian LEZAMA CUELLAR
Secretario del Proceso
Departamento Académico de Matemática y Física



C.c.:
Bach. ELMER TUEROS PUCHURI
Jurados (4)
Archivo



UNSCH

FACULTAD DE
INGENIERÍA
DE MINAS, GEOLOGÍA Y CIVIL



“Año de la unidad, la paz y el desarrollo”

CONSTANCIA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN

CONSTANCIA N° 071-2023-FIMGC

El que suscribe; responsable verificador de originalidad de trabajos de tesis de pregrado con el software Turnitin, en segunda instancia para las **Escuelas Profesionales** de la **Facultad de Ingeniería de Minas, Geología y Civil**; en cumplimiento a la **Resolución de Consejo Universitario N° 039-2021-UNSCH-CU**, Reglamento de Originalidad de Trabajos de Investigación de la Universidad Nacional San Cristóbal de Huamanga y **Resolución Decanal N° 288-2023-FIMGC- UNSCH-D**, deja constancia de originalidad de trabajo de investigación, que el/la Sr./Srta.

Apellidos y Nombres : TUEROS PUCHURI, ELMER
Escuela Profesional : INGENIERÍA DE SISTEMAS
Título de la Tesis : “APLICACIÓN WEB PARA CONTROL DE ASISTENCIA DE ESTUDIANTES EN EL CENTRO PREUNIVERSITARIO DE LA UNSCH, REGIÓN AYACUCHO, 2023”
Evaluación de la Originalidad : 28 % Índice de Similitud
Identificador de la entrega : 2159580287

Por tanto, según los Artículos 12, 13 y 17 del Reglamento de Originalidad de Trabajos de Investigación, es **PROCEDENTE** otorgar la **Constancia de Originalidad** para los fines que crea conveniente.

En señal de conformidad y verificación se firma la presente constancia

Ayacucho, 06 de setiembre del 2023



UNIVERSIDAD NACIONAL DE
SAN CRISTÓBAL DE HUAMANGA
Facultad de Ingeniería de Minas, Geología y Civil

Mg. Ing. Christian LEZAMA CUELLAR
Verificador de Originalidad de Trabajos de Tesis de Pregrado
Departamento Académicos de Matemática y Física



Con depósito para Sustentación y Tramites
Cc. Archivo

FACULTAD DE INGENIERIA DE MINAS, GEOLOGIA Y CIVIL
Av. Independencia S/N Ciudad Universitaria
Central Tel. 066 312510
Anexo 151

APLICACIÓN WEB PARA CONTROL DE ASISTENCIA DE ESTUDIANTES EN EL CENTRO PREUNIVERSITARIO DE LA UNSCH, REGIÓN AYACUCHO, 2023

INFORME DE ORIGINALIDAD

28%

INDICE DE SIMILITUD

18%

FUENTES DE INTERNET

1%

PUBLICACIONES

27%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

| | | |
|---|--|-----|
| 1 | Submitted to Webster University Trabajo del estudiante | 13% |
| 2 | Submitted to Universidad Nacional de San Cristóbal de Huamanga Trabajo del estudiante | 8% |
| 3 | repositorio.unsch.edu.pe Fuente de Internet | 5% |
| 4 | repository.udistrital.edu.co Fuente de Internet | 1% |
| 5 | hdl.handle.net Fuente de Internet | <1% |
| 6 | repositorio.unan.edu.ni Fuente de Internet | <1% |
| 7 | www.pancanal.com Fuente de Internet | <1% |
| 8 | repositorio.upsin.edu.mx Fuente de Internet | <1% |

9

repositorio.ucv.edu.pe

Fuente de Internet

<1 %

10

repositorio.unp.edu.pe

Fuente de Internet

<1 %

Excluir citas

Activo

Excluir coincidencias < 30 words

Excluir bibliografía

Activo