

DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN OPERATIVA,
FUNCIONAL Y ADMINISTRATIVA DE LOS HOGARES JUVENILES UNIVERSITARIOS

ANDRÉS GARCIA ATEHORTUA
BRAYAN STIVEN ERAZO SERNA

UNIVERSIDAD CATÓLICA DE ORIENTE
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
RIONEGRO

2020

DESARROLLO DE UNA PLATAFORMA WEB PARA LA GESTIÓN OPERATIVA,
FUNCIONAL Y ADMINISTRATIVA DE LOS HOGARES JUVENILES UNIVERSITARIOS

ANDRÉS GARCIA ATEHORTUA
BRAYAN STIVEN ERAZO SERNA

Trabajo para obtener el título de ingeniero de Sistemas

Asesores

ALEJANDRO RAMIREZ
MARIA VICTORIA SILVA

Universidad Católica de Oriente

Facultad de Ingenierías

Programa de Ingeniería de Sistemas

Rionegro

2020

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Dedicatoria:

A la universidad Católica de Oriente por haber sido nuestra alma mater y lugar de formación por tantos años.

Al hogar Santa Maria por ser el lugar de residencia de varios años para nosotros durante el transcurso de nuestros estudios.

Agradecimientos:

A nuestros padres que nos apoyaron y financiaron mayoritariamente durante todo el transcurso de nuestros estudios.

A la universidad y sus profesores que cumplieron la gran tarea de transmitir sus conocimientos durante los semestres de manera respetuosa y colaborativa.

Al Hogar Santa Maria de la UCO que nos brindó posada y alimentación durante parte de nuestros estudios, además del servicio y guía constante de sus trabajadores.

Contenido

1. Antecedentes:	7
2. Planteamiento del problema:	9
3. Justificación:	12
4. Objetivos:	14
4.1. General:	14
4.2. Específicos:	14
5. Marco teórico	15
6. Diseño metodológico	19
7. Resultados	25
7.1. Objetivo General	25
7.2. Objetivo Específico 1	26
7.3. Objetivo Específico 2	27
7.4. Objetivo Específico 3	29
7.5. Objetivo Específico 4	30
8. Conclusiones	32
9. Referencias bibliográficas	35
10. Apéndices	38

Antecedentes

Durante la fase de consulta e investigación no encontramos nada relacionado con la gestión y organización por medio de plataformas digitales en los hogares juveniles universitarios; por lo tanto, consideramos que este es un trabajo pionero en ese ámbito.

Los hogares universitarios son una estrategia para acercar el conocimiento a todos los lugares y personas del país, debido que no todas las personas cuentan con un alojamiento cerca de una institución a la que quieren ir a estudiar; lo que hace que el costo de su transporte sea muy alto y que los tiempos de desplazamiento no sean óptimos, para este problema existen los hogares juveniles, los cuales facilitan el hospedaje mientras las personas se encuentren estudiando cualquier tipo de formación profesional, estos normalmente se hacen conocer por las universidades o por las personas que conocen sobre estos servicios e inclusive redes sociales, porque si tomamos en consideración los hogares estudiantiles del Oriente antioqueño se puede observar que estos hogares no cuentan con una página web.

A continuación, se describen algunos hogares universitarios que se encuentran en el Oriente antioqueño:

Hogar Juvenil Universitario Santa María según la Udea “fue inaugurado por la Universidad Católica de Oriente en 2012, “fue una idea del Departamento de Bienestar Universitario con el objetivo de brindar a los estudiantes de las partes lejanas del área de influencia de la Universidad, una vivienda con recursos más favorables y así completar sus estudios”, cuenta Elizabelt Quiceno, administradora del Hogar” localizada en el Carmen-Aguas claras, vía Rionegro-La ceja, no cuenta con una portal web, este lugar se ha dado a conocer virtualmente gracias a las universidades como la UCO y la UdeA.

Hogar juvenil universitario filipense localizado en Rionegro- Antioquia Calle 51 N° 47-40 Barrio San Francisco, no cuenta con una página web que dé a conocer este lugar, aunque hay que tener en cuenta que este hogar hace parte de la Congregación de RR. Filipenses Misioneras de Enseñanza que cumple un fin que es según la página oficial Filipenses de Enseñanza “El fin específico de la Congregación es la «Renovación cristiana de la Sociedad» de acuerdo con las necesidades más urgentes de la Iglesia y con especial atención a los más necesitados. Lleva a cabo esta misión a través de obras de promoción humana y educación cristiana en Colegios, Residencias y Casas de Espiritualidad; colaboración en Parroquias y ayuda a movimientos eclesiales.” teniendo en cuenta lo anterior se da gracias a esta finalidad, que este hogar o lugar de residencia para estudiantes se haya realizado.

Planteamiento del problema

El hogar juvenil universitario Santa María cuenta con los espacios y recursos necesarios para que los estudiantes puedan tener un mejor enfoque en su formación académica, para el ingreso de un estudiante al hogar Juvenil se tomará como referencia el proceso que aplica el hogar de la UCO o también conocido como Hogar juvenil Universitario Santa María, en este el solicitado deberá entregar un formulario con información personal y de su familia al administrador del hogar, para que se pueda hacer un estudio de la solicitud y así aprobar o no la solicitud; también se debe entregar toda la información relacionada del estudio que se está realizando por parte del estudiante como: el costo de matrícula, semestres cursados, asignaturas elegidas y horarios del semestre actual, en el caso que ya haya cursado un semestre, el promedio debe superar a 3.8 para cumplir con los requisitos del hogar.

A la hora de asignar habitaciones y actividades de aseo para los estudiantes, el administrador lo realiza de una manera aleatoria, prácticamente consta de ingresar papelitos con las habitaciones y actividades en una bolsa para que los estudiantes saquen un papelito con sus correspondientes habitaciones y actividades de aseo.

El costo mensual de la estadía en el hogar es de 250 mil pesos, se da la oportunidad de pagar mensualmente o al final del semestre, el estudiante debe quedar paz y salvo, debido a unos días adicionales que son contados por el administrador se debe pagar 1.100.000 pesos para el semestre completo, menor que el costo por los mismos servicios de manera externa que se pueden encontrar en el municipio de Rionegro en el mismo intervalo de tiempo.

En estos hogares por lo general se tiene un lugar para vivir que puede ser una habitación individual o compartida dependiendo del caso, una alimentación adecuada dividida en tres comidas al día las cuales cada una tiene un horario y una franja horaria en la cual está disponible, a cambio de realizar unas actividades internas propias de

cada hogar cómo lo puede ser el lavado de la loza de alguno de los turnos de la alimentación, ayuda para la preparación de los alimentos, y además de tener una actividad de aseo determinada que debe ser realizada a una área general, además del aseo de cada bloque y/o habitación en específico previamente definida; este Hogar también cuenta con una lavadora y una secadora que está a disposición de los habitantes del hogar, además de áreas de acceso común, ya sea para momentos de esparcimiento o de estudio, como espacios libres o salones de estudios, sala de TV y sala de juegos, la cual cuenta con una mesa de ping pong y una cafetera; los estudiantes también cuentan con el uso de una planilla para registrar la salida y la entrada de actividades ajenas a los horarios de clase, cómo compromisos laborales, personales y extra clases.

El servicio de acceso común, como los salones o espacios libres están disponibles desde las 6 am hasta las 10:30 pm, la lavandería está disponible para todo el personal desde las 6 am hasta las 7 pm cualquier día de la semana.

En este hogar como en cualquier otro, para un buen funcionamiento o una buena convivencia deben existir reglas o normas que permitan una gestión óptima para un hogar, en el caso del hogar Santa María se tienen algunas normas como:

- horarios específicos para la alimentación
- Todos dentro de sus respectivos bloques antes de las 11 pm.
- Respetar el sueño de los demás haciendo silencio desde las 11 pm.
- Horarios de entrada los fines de semanas hasta las 10 pm y en semana hasta las 10:30 pm.
- No consumir alcohol o estupefacientes.
- Promedio del semestre anterior superior a 3.8.

Todas estas actividades y características son analizadas y observadas tomando como referencia el Hogar Juvenil Universitario Santa María de la UCO durante 2019-1, 2019-2 y 2020-1.

Tomando como referencia el Hogar Filipino, este cumple con actividades similares, pero cuenta con personal de servicio de aseo y las actividades que deben hacer por parte del estudiante es menor al del hogar Santa María, estos estudiantes tienen un día específico para realizar el aseo de su correspondiente bloque, por lo general es el sábado aunque se van turnando para realizar estos aseos y adicionalmente los estudiantes que deseen participar en un descuento en la mensualidad del hogar pueden pagar 5 horas semanales realizando la labor de portero o ayudando a lavar la loza, esta mensualidad consta de un valor de 400.000 pesos, que con el descuento respectivo queda en 370.000 pesos, también cuenta con las alimentaciones correspondientes a desayuno, almuerzo y cena con un horario determinado para esto de 7 a 8am, de 11:00am a 12:30m y de 6pm a 7pm correspondientemente, también cuenta con una lavadora que está disponible los martes, jueves y sábados para los estudiantes y estos deben registrar los horarios en el que la van a utilizar por medio de un formato o planilla.

Todas estas actividades y características son analizadas y observadas tomando como referencia el Hogar Filipino durante 2019-1 y 2019-2.

Justificación

El desarrollo de esta Plataforma Web es un reto debido a que plantea mejorar el funcionamiento interno de un hogar universitario, estos funcionan de manera muy centralizada por lo que casi toda la responsabilidad cae sobre los administradores, esto combinado con una población bastante grande que sobrepasa las capacidades de tales personas encargadas; desde el punto de vista social es importante debido a que al mejorar las condiciones del administrador este actuaría de una manera más eficiente, esto se vería reflejado en un incremento de la efectividad de sus acciones; esto también aplica para los estudiantes ya que al tener una herramienta para gestionar sus actividades pendientes en el hogar se podrán programar mejor para cumplir con estas, como el turno de lavado de loza y el aseo de la área común en particular, esto les permite equilibrar sus responsabilidades con el hogar y cumplir con la respectiva carga académica para que ninguna de las dos se vea afectada negativamente por la otra; otro aspecto a tener en cuenta es de la convivencia entre los mismos estudiantes que se verá mejorada al mejorar la comunicación con el administrador, evitando falencias en la comunicación que se pueden presentar cuando se busca a alguna persona por cierto motivo y esta no se encuentra, por lo que se pierde tiempo en realizar esta tarea que se puede haber evitado.

Para el área funcional es conveniente ya que si todas las áreas comunes están listas para ser usadas significa que la mayoría están cumpliendo con sus actividades designadas, esto evitará problemas de convivencia de este tipo entre los propios estudiantes al evitar cuestionarse quién es el encargado del aseo de ciertas áreas, además de evitar llamados de atención innecesarios por parte del administrador al haber incumplido con cierta tarea sin importar el motivo, hay que dejar en claro que en algunas situaciones hay excepciones, por ejemplo si el estudiante se encuentra enfermo o en alguna salida de campo o entre otras, entonces se toma en consideración estos casos y deberán ser manejados estrictamente por el administrador y el estudiante.

Para el área operativa es conveniente ya que el tiempo que toma realizar algunas acciones se verá reducido considerablemente, cómo llenar la planilla para la salida, ya no habría que buscarla ni llenarla, solo sería especificar el motivo de la salida y hora de llegada y el lugar de destino; También tener en cuenta el uso y las reservaciones de los salones, revisar el horario de un estudiante ya no dependerá de la memoria, solo sería abrir la aplicación y seleccionar al estudiante para que nos muestre toda su información, si ha salido, que turno de loza le toca, que aseo general debe hacer y si lo ha hecho o no, todo al alcance de la mano, esto reduce los tiempos de respuesta entre el administrador y estudiantes al no tener un encuentro presencial para preguntas tan elementales como si ya realizó cierta actividad.

Para el aspecto económico se puede reducir el consumo de papel que se requiere para las planillas.

Estos tipos de servicios sociales que se brindan para ayudar a los estudiantes son una excelente forma de apoyar el avance de los estudiantes con sus estudios de educación superior, pero estos también requieren un compromiso por parte del estudiante, este compromiso varía de acuerdo al hogar en el que se encuentren, pero en general se requiere hacer como mínimo una actividad de aseo en un área de uso común y en un horario establecido.

Ya que la carga académica puede llegar a ser muy exigente para los estudiantes, y estos sin mala intención pueden llegar a olvidar la realización de sus compromisos, por lo que disponer de una herramienta para planificar y organizar todos los compromisos de un estudiante con su lugar de residencia, facilitará el cumplimiento de estos con tales actividades, además de facilitar y mejorar el seguimiento por parte de la administración.

Objetivos

General

Desarrollar la plataforma web para la gestión Operativa, Funcional y Administrativa en los hogares juveniles universitarios.

Específicos

- Analizar la situación problemática y hacer el levantamiento de requerimientos.
- Decidir la arquitectura más adecuada para usar según el análisis del problema.
- Selección de las tecnologías y técnicas de trabajo.
- Implementación de la plataforma web.

Marco teórico

Software: conjunto de instrucciones programadas que una computadora debe seguir, hay varios tipos, pero en este caso nos interesa el software de aplicaciones que es aquel que nos ayuda y facilita una tarea o conjunto de tareas en específico, permitiéndonos ser más productivos y eficaces.

Arquitectura de software: Son patrones o lineamientos que ayudan a la construcción de un programa, esta guía ayuda a todo el conjunto de personas relacionadas con la aplicación como desarrolladores y analistas, de manera que esta pueda cumplir todos sus requerimientos.

Paradigmas: Un paradigma en programación es aquella propuesta adoptada que se usa para solucionar un problema, existen muchos tipos de estos pero cada uno resuelve problemas específicos mejor o peor que los demás; por esta razón es importante considerarlos a la hora de plantear soluciones al problema, los más conocidos o usados suelen ser:

- Funcional: los programas se componen de funciones, que reciben una entrada y devuelve una salida.
- Orientado a Objetos: se programa en base a entidades que representan elementos de nuestra realidad problemática, con sus atributos y comportamientos.
- Dirigido a Eventos: El flujo del programa está determinado por sucesos externos, como pueden ser las acciones de un usuario.

Requisitos: En la ingeniería de software es el proceso de recoger, analizar y documentar las necesidades y problemas del cliente, que sirven para crear la aplicación en concordancia con lo que el cliente necesita y espera, generalmente consta de cuatro etapas:

- Identificación: se habla con el cliente y se analizan las necesidades a resolver.
- organización: de acuerdo al criterio escogido, puede ser importancia, orden, o otros.
- Negociación: los requisitos deben ser modificados para evitar la ambigüedad.
- Documentación: se documentan para tenerlos de manera tangible.

Requerimiento: Son las especificaciones técnicas, funcionales, visuales, e impuestas por el usuario que la aplicación debe cumplir para lograr su objetivo; cómo pueden ser que tenga un diseño en específico o que se utilice cierta tecnología; suelen dividirse en:

- Requerimientos funcionales: son las prestaciones o funciones que debe cumplir la aplicación.
- Requerimientos NO Funcionales: son asociados generalmente con características como seguridad, rendimiento y usabilidad del software

Buenas prácticas de desarrollo: Es una técnica o método que se usa para garantizar la satisfacción del cliente y la confiabilidad de aplicación, pueden referirse a la toma correcta de decisiones, mantener presupuestos y metas de manera realista, optimización y análisis del sistema, codificación y validación rigurosa del código, documentación y control de calidad; cada una destinada a mejorar y fortalecer alguno de los muchos aspectos de la aplicación; algunas de las más utilizadas e importantes suelen ser:

- implementación modular del código.

- código autodocumentado.
- estándares de nombramiento.
- reutilización de código.
- uso de patrones.
- definición de responsabilidades únicas.

POO: Paradigma de programación que data de los años 80 para mejorar la gestión del código, aumentar la mantenibilidad y modificabilidad de la aplicación, además de muchos más beneficios.

DOO: El Diseño Orientado a Objetos toma todos los componentes de POO y los implementa de manera particular con un lenguaje de programación, cabe resaltar que el sistema a nivel analítico y conceptual es totalmente independiente del lenguaje en que se implemente.

Servidor: En el contexto de la informática hace referencia a un Hardware o Software que expone recursos para ser consumidos, a nivel físico se trataría de una máquina integrada a una red ya sea interna o conectada a internet.

Plataforma web: Es un sitio web que nos permite realizar varias tareas bajo el mismo ambiente, con solamente una conexión a internet para que el usuario pueda acceder.

Plataforma responsive: Es una técnica o diseño para realizar plataformas en el cual se adaptan a diferentes tipos de dispositivos, por lo general se busca que la apariencia de una plataforma web o app se adapte a los diferentes tamaños de pantalla que tienen los dispositivos.

Android: Es un sistema operativo hecho para móviles, basado en Linux, se puede programar con Java o más recientemente Kotlin, debido a su integración con los servicios de Google y otras características como ser gratuito y su alta capacidad de personalización es el sistema operativo para celulares con más cuota de mercado actualmente.

IOS: Es el segundo sistema operativo para dispositivos móviles con más cuota de mercado, propiedad de Apple y exclusivo para los productos de esta compañía.

Framework: Es una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

SPA: Es una aplicación web de una sola página, donde todos los archivos de HTML, CSS y JS se cargan de una sola vez, dando una experiencia de navegación normal sin tener que recargar el navegador o cambiar de una página a otra, solo se van intercambiando entre vistas.

HTML: Es un lenguaje de marcado de hipertexto que se utiliza para el desarrollo de páginas de Internet, es un lenguaje que interpreta el navegador web para mostrar los sitios o aplicaciones web tal y como estamos acostumbrados.

CSS: Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

Angular: Es un framework que facilita la creación de páginas web.

Bootstrap: Es una librería que combina CSS, HTML y JS para desarrollar front-end de un modo más sencillo y rápido.

React: Librería de código abierto de Facebook para la construcción de interfaces de usuario.

ionic: es un framework diseñado para crear aplicaciones híbridas para Android e iOS; además de ser compatible con Angular y React lo que facilita mucho el desarrollo web; es una herramienta de trabajo que también es bastante popular por lo que tiene una gran comunidad de desarrolladores que la mantienen vigente y una gran documentación técnica.

Bases de Datos: El propósito de las bases de datos surge con la necesidad de registrar y almacenar datos para poderlos consultar fácil y velozmente; generalmente existen de 3 tipos:

- Relacionales: se basan en una estructura planificada para almacenar los datos y usan SQL para las diversas operaciones generalmente conocidas como CRUD, del inglés (Create, Read, Update, Delete).
- No-Relacionales: Estas son más flexibles en cuanto a consistencia de datos y se han convertido en una opción que intenta solucionar algunas limitaciones que tiene el modelo relacional.
- No-tradicionales: son aquellas diferentes de las 2 anteriores, como los árboles o los grafos.

ORM: Mapeo Objeto-Relacional que permite convertir los objetos (mapeo) y guardar la información en una base de datos (persistencia); evitando lo laborioso que es transformar toda la información que se recibe de la base de datos, en los objetos de la aplicación y viceversa. Utilizando un ORM este mapeo será automático e independiente de la base de datos que se esté utilizando, pudiendo cambiar de motor en cualquier momento.

JSON: Es un formato basado en texto estándar para representar datos estructurados en la sintaxis de objetos de JavaScript. comúnmente utilizado para transmitir datos en aplicaciones web, por ejemplo: enviar datos desde el servidor al cliente, para ser utilizados en páginas web, o viceversa.

API: Es un conjunto de reglas (código) y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas; sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software; son valiosas porque permiten hacer uso de funciones ya existentes en otro software (o de la infraestructura ya existente en otras plataformas) para no reinventar la rueda, reutilizando código probado y que funciona correctamente.

PRUEBAS: Es la realización de pruebas sobre el software, con el fin de obtener información acerca de su calidad, este proceso tiene varios objetivos bien definidos, como son:

- Encontrar defectos o bugs con el fin de ser subsanados permitiendo aumentar la calidad del software.
- Corregir y subsanar errores, aumentamos nuestra confianza en el mismo.
- Evitar la aparición de nuevos defectos, ya que si aumento la calidad de mi software y se hacen las cosas bien, es menos probable que aparezcan nuevos defectos a la larga.

JWT: Es un estándar abierto que define una forma compacta y autónoma de transmitir información de forma segura entre las partes como un objeto JSON. Esta información se puede verificar y confiar porque está firmada digitalmente. Los JWT se pueden firmar

usando una clave secreta, o un par de claves pública / privada cifrando el contenido; generalmente usados para la autenticación de usuarios, con un solo inicio de sesión; cada solicitud posterior incluirá el JWT, lo que permitirá al usuario acceder a rutas, servicios y recursos que están permitidos con ese token.

CLI: La interfaz de línea de comandos es un programa que permite a los usuarios escribir comandos de texto instruyendo a la computadora o software para que realice alguna tarea en específico; requiere de muy pocos recursos por lo que es ideal para entornos de alto rendimiento y de mucha precisión.

HTTP: Es un protocolo que nos permite realizar una petición de datos y recursos, basado en el principio de cliente-servidor: las peticiones son enviadas por una entidad: el agente del usuario. La mayoría de las veces es un navegador Web, Cada petición individual se envía a un servidor, el cual la gestiona y responde, negando o devolviendo el recurso que se solicitó.

REST: Es cualquier interfaz de intercambio de datos entre sistemas que use HTTP para obtener o generar operaciones sobre esos datos, en cualquiera de los formatos XML o JSON; este tipo de interfaces son ampliamente usadas en la industria del software.

Control De Versiones: Es una herramienta de software que monitoriza y gestiona cambios en un sistema de archivos, también ofrece herramientas de colaboración para compartir e integrar dichos cambios con otros usuarios. Al operar al nivel del sistema de archivos, monitorizará las acciones de adición, eliminación y modificación aplicadas a archivos y directorios.

Anotaciones: Es una manera de incrustar información suplementaria en un archivo fuente, Esta información, no cambia las acciones de un programa; sin embargo, esta metadata puede ser utilizada por varias herramientas, durante tiempo de compilación cómo de ejecución; cabe aclarar que en tecnologías web cómo Angular esto también existe, pero son llamados decoradores.

IDE: Es un software para el desarrollo de aplicaciones que combina herramientas del desarrollador comunes en una sola interfaz gráfica de usuario, tales cómo, depurador de código, editor de código fuente, compilador, enlace con bases de datos y muchas más funciones, todas en un mismo software.

Gestión de Dependencias: Las dependencias son las herramientas, plugins, librerías y frameworks necesarios para desarrollar aplicaciones web de alto nivel, por lo que el manejo de estas es fundamental para el debido funcionamiento de un proyecto; estas herramientas permiten mantenerlas organizadas, y sobre todo, actualizadas.

Principios: Son comportamientos, técnicas y consejos que un buen programador debería tener en cuenta siempre para cualquier desarrollo, no siempre aplican y en ocasiones pueden chocar entre sí, pero es imperativo cómo mínimo conocerlos, los más famosos son:

- **SOLID:** Bastantes conocidos y prácticamente un estándar a seguir en POO.
- **KISS:** La simplicidad sobre todo, evitar la complejidad lo más posible.
- **DRY:** Entre menos código se repita mejor, la reutilización es fundamental.

Patrones: Son soluciones para problemas que se presentan de manera común en muchos desarrollos, estas soluciones han sido probadas e implementadas con anterioridad, por lo que al adaptarlo a cualquier proyecto el problema está casi solucionado; también es necesario tener cuidado con ellos ya que implementarlos de manera errónea o injustificada trae problemas más difíciles de solucionar.

Hogar juvenil Universitario: Lugar donde pueden vivir y estudiar las personas que estén cursando un estudio de educación superior, puede ser tecnológico, pregrado o posgrado.

Administrador: Persona encargada de gestionar y organizar un lugar o a un personal en específico.

Convivencia: Hace referencia a las relaciones interpersonales que tiene cierta población con sus contemporáneos, superiores o subordinados.

Diseño metodológico

FASE-1: Analizar la situación problemática y hacer el levantamiento de requerimientos.

ETAPA-1: Análisis del funcionamiento del Hogar Juvenil Universitario Santa María.

ACTIVIDAD-1: Realizar una entrevista con el administrador del hogar para saber el funcionamiento interno del hogar Santa María.

TAREA-1: Contactar con el administrador para asignar una fecha de encuentro.

TAREA-2: Recopilar todas las actividades y particularidades del hogar Santa María durante la entrevista con el administrador.

ETAPA-2: Definir qué actividades son problemáticas debido a factores de incumplimiento o eficacia.

ACTIVIDAD-1: Discernir las tareas y compromisos que se realizan en el hogar por parte de los estudiantes.

TAREA-1: Crear una lista de funciones a implementar en la aplicación.

ACTIVIDAD-2: Diferir cuáles son las actividades que pueden ser optimizadas dentro del hogar.

TAREA-1: Preguntar al administrador información adicional del acontecer de cada una de las actividades en el hogar para saber su funcionamiento preciso.

TAREA-2: Analizar qué tareas pueden ser optimizadas en tiempo o dinero.

ETAPA-3: Decidir las funciones a implementar después de haber analizado las necesidades y de haber listado todas las posibles funciones.

ACTIVIDAD-1: Definir qué funciones serán implementadas y dar una breve explicación del porqué.

TAREA-1: Demostrar la utilidad de cada una de las funciones que se van a implementar.

TAREA-2: Explicar por qué no se van a implementar las otras posibles funciones.

ACTIVIDAD-2: Explicar cómo planean resolver alguno o varios de los subproblemas previamente identificados.

TAREA-1: Aclarar cómo se planea que cada función resuelva o mejore la situación de cada uno o varios de los subproblemas identificados.

ETAPA-4: Hacer aproximaciones de costos y tiempos requeridos para el desarrollo de la aplicación

ACTIVIDAD-1: Considerar que es un desarrollo gratuito, por lo que la reducción de los costos es una de las prioridades

TAREA-1: Proyectar un cronograma con la menor duración de tiempo posible.

TAREA-2: Intentar que todas las tecnologías y herramientas a usar sean código abierto y/o gratuitas.

FASE-2: Decidir la arquitectura más adecuada para usar según el análisis del problema.

ETAPA-1: Consultar o profundizar diferentes tipos de arquitecturas.

ACTIVIDAD-1: Definir cuales arquitecturas se ya conocen y consultar otros tipos de arquitecturas.

TAREA-1: Definir qué arquitecturas conocemos a profundidad que no sea necesario consultar acerca de ellas.

TAREA-2: Consultar diferentes tipos de arquitecturas que se relacionen con la programación móvil y/o programación web.

ACTIVIDAD-2: Hacer un análisis inicial de que arquitecturas pueden ser usadas para programación web o programación de aplicaciones móviles.

TAREA-1: Diferir cuales arquitecturas están orientadas a servicios que puedan ser consumidos fácilmente desde cualquier dispositivo.

ACTIVIDAD-3: Evaluar todas las arquitecturas para saber cuándo se usan y qué situaciones o problemas son resueltos fácilmente por alguna arquitectura en específico.

TAREA-1: Preseleccionar arquitecturas que puedan llegar a ser usadas, basándose en los casos que son buenas y que sirvan para el ambiente web y/o móvil

ETAPA-2: Analizar nuestros requisitos funcionales y no funcionales para considerar que arquitecturas preseleccionadas con anterioridad nos pueden satisfacer o no.

ACTIVIDAD-1: Decidir qué requisitos funcionales están directamente relacionados con la arquitectura.

ACTIVIDAD-2: Analizar si alguno de los requisitos NO funcionales está relacionado con la arquitectura

ETAPA-3: Deliberar cuál de las arquitecturas preseleccionadas usar.

ACTIVIDAD-1: Decidir qué arquitectura usar.

TAREA-1: Consultar información adicional de la arquitectura seleccionada; temas acerca de su implementación.

ACTIVIDAD-2: Empezar a diseñar nuestra plataforma web basándonos en la arquitectura seleccionada.

FASE-3: Selección de las tecnologías y técnicas de trabajo.

ETAPA-1: Consultar y/o profundizar acerca de marcos de trabajo en el desarrollo de software.

ACTIVIDAD-1: Consultar y/o profundizar acerca de marcos de trabajo ágiles y tradicionales.

TAREA-1: Aclarar y/o profundizar acerca de marcos de trabajo Ágiles

TAREA-2: Aclarar y/o profundizar acerca de marcos de trabajo tradicionales.

ACTIVIDAD-2: Determinar las características principales de cada uno de los marcos de trabajo consultados.

TAREA-1: Determinar las características principales de cada uno de los marcos de trabajo ágiles que se consulten o que sepamos con antelación.

TAREA-2: Determinar las características principales de cada uno de los marcos de trabajo tradicionales que se consulten o que sepamos con antelación.

ACTIVIDAD-3: Deliberar qué marco de trabajo de todos los consultados previamente se adapta mejor a nuestra situación.

ACTIVIDAD-4: Definir un marco de trabajo para realizar nuestra aplicación.

ETAPA-2: Seleccionar el lenguaje de programación a utilizar en el Backend.

ACTIVIDAD-1: Profundizar y/o consultar lenguajes de POO además de java.

TAREA-1: Deliberar si usar JAVA o aprender otro lenguaje de POO.

TAREA-2: Definir a JAVA como el lenguaje a usar para nuestro Backend.

ETAPA-3: Seleccionar el framework y/o librerías para usar en la parte de programación web.

ACTIVIDAD-1: Consultar y/o profundizar acerca de Angular y React.

TAREA-1: Profundizar acerca del manejo de Angular para el desarrollo de SPA.

TAREA-2: Profundizar acerca del manejo de Angular y el uso del patrón MVC.

TAREA-3: Consultar acerca de React y su manejo de componentes y estados.

TAREA-4: Consultar acerca de React sobre los archivos JSX y enlace de datos.

ACTIVIDAD-2: Definir pros y contras de cada uno de los 2, de acuerdo a nuestro proyecto y las características de cada tecnología.

TAREA-1: Sopesar los pros y contras de cada uno de los 2.

ACTIVIDAD-3: Seleccionar alguno de los 2 para crear nuestra capa de presentación.

ACTIVIDAD-4: Integración de la librería de Bootstrap para la visualización de la página web.

ETAPA-4: Seleccionar el framework y/o lenguajes de programación para usar en la parte de programación móvil.

ACTIVIDAD-1: consultar acerca de los frameworks de Ionic y Flutter.

TAREA-1: Consultar acerca de las características principales de Ionic.

TAREA-2: Consultar sus integraciones con bases de datos.

TAREA-3: Consultar su manejo y/o compatibilidad con Angular y React.

TAREA-4: Consultar acerca de las características principales de Flutter.

TAREA-5: Consultar acerca del lenguaje Dart.

TAREA-6: Consultar acerca de SQLite.

ACTIVIDAD-2: Definir pros y contras de cada uno de los 2, de acuerdo a nuestro proyecto y las características de cada tecnología.

TAREA-1: Sopesar los pros y contras de cada uno de los 2.

ACTIVIDAD-3: Seleccionar alguno de los 2 para crear la aplicación móvil.

FASE-4: Implementación de la plataforma web

ETAPA-1: Aprender Ionic o Flutter, dependiendo de la seleccionada para crear nuestra aplicación móvil.

ACTIVIDAD-1: buscar contenido y cursos de calidad para empezar a aprender el framework que se escoja.

ETAPA-2: Desarrollo del Backend de la aplicación.

ACTIVIDAD-2: Crear la implementación del código con todas las funciones que se vayan a implementar y teniendo en cuenta los requerimientos del hogar.

TAREA-1: Hacer uso de buenas prácticas de programación y de los estándares de nombramiento de JAVA.

TAREA-2: Definir las capas de la aplicación de manera clara.

TAREA-3: Realización de pruebas unitarias y de integración a nuestro código.

TAREA-4: Implementación y prueba de una o conjunto de APIs que puedan ser consumidas desde cualquier dispositivo.

ETAPA-3: Desarrollo de la capa de presentación de la aplicación.

ACTIVIDAD-1: Integración y uso de Bootstrap para la capa de presentación de la aplicación.

ACTIVIDAD-2: Fortalecer los conocimientos de HTML, CSS y JS.

TAREA-1: Diseño y Desarrollo de la aplicación web basándonos en las características que nos ofrece Bootstrap cómo el uso del grid y las clases container.

FASE-5: Implementación de JWT y añadir medidas de seguridad informática a la aplicación

ETAPA-1: Consultar acerca de JWT.

ACTIVIDAD-1: Buscar contenido y ejemplos de implementaciones en JAVA.

TAREA-1: Siguiendo el tutorial que se haya escogido implementar el servidor de JWT en el Backend.

ETAPA-2: Consultar otras posibles medidas de seguridad cómo los CORS.

ACTIVIDAD-2: Crear una configuración personalizada de estos.

TAREA-1: Filtrar todas las peticiones que ingresen al Backend verificando que contengan un token JWT válido.

TAREA-2: Filtrar todas las peticiones que ingresen al Backend verificando que cumplan el protocolo HTTP.

ETAPA-3: Consultar cómo cifrar información sensible

ACTIVIDAD-1: Buscar algoritmos o librerías para cifrar información sensible cómo las contraseñas.

ACTIVIDAD-2: Implementación de métodos para cifrar y descifrar la información en el Backend.

TAREA-1: Guardar las contraseñas cifradas en la base de datos.

ETAPA-4: Adaptar la aplicación web para el uso de Tokens.

ACTIVIDAD-1: Buscar la manera de guardar información en los dispositivos de los usuarios,

ACTIVIDAD-2: Implementación de métodos para guardar, modificar, consultar y eliminar la información que esté almacenada en los dispositivos del cliente.

TAREA-1: Añadir el token a cada una de las solicitudes que la aplicación web haga al backend.

ETAPA-5: Adaptar la aplicación móvil para el uso de Tokens.

ACTIVIDAD-1: Buscar la manera de guardar información en los dispositivos móviles de los usuarios,

ACTIVIDAD-2: Implementación de métodos para guardar, modificar, consultar y eliminar la información que esté almacenada en los dispositivos del cliente.

TAREA-1: Añadir el token a cada una de las solicitudes que la aplicación móvil haga al backend.

Resultados

Objetivo General

El objetivo principal de desarrollar la plataforma para la gestión del Hogar Universitario Juevenil Santa Maria se cumplió en la mayoría de las funciones propuestas que fueron especificadas anteriormente, y otras pocas de manera parcial; también los tres frentes de acción fueron completados a cabalidad, los cuales fueron:

- La página web, diseñada principalmente para el uso de los administradores.
- La aplicación móvil, diseñada principalmente para el uso de los estudiantes.
- La lógica del negocio, diseñada para ser usada por todos los usuarios.

Objetivo específico I

Este objetivo específico corresponde a la etapa de Levantamiento de requisitos y requerimientos en el desarrollo de software.

El levantamiento de requerimientos y el análisis de la situación problemática fue el primer objetivo cumplido a cabalidad, además de haber sido el más fácil de realizar debido a que los realizadores habían vivido en el Hogar Santa Maria previamente, dotándolos de un conocimiento previo bastante valioso acerca del funcionamiento interno del hogar.

La entrevista que se realizó con el administrador del hogar Santa Maria fue realizada a finales de noviembre de 2019, el contenido de la misma no fue guardado ya que fue una entrevista de tipo informal, con el objetivo de informar más que negociar los requerimientos y requisitos de la plataforma, esto debido a que el administrador se le había informado con anterioridad la idea de este proyecto, logrando que el planteamiento en sí fuera apoyado por el mismo, proveyendo necesidades específicas cómo administrador y brindando nuevas perspectivas que no estaban a nuestro alcance.

Este hecho también permitió una interacción más oportuna y rápida con el administrador de turno por lo que el levantamiento de requerimientos fue realizado rápida y eficazmente; durante la charla con el administrador éste aprobó todas las funciones propuestas y solo pidió un requerimiento en específico que fue la adición de la firma.

El contenido de esta entrevista fue bastante informal en cuanto al ámbito técnico por lo que el administrador entendió la idea general del proyecto, y estuvo de acuerdo en que la aplicación para celulares fuera solo para estudiantes y que la página web fuera para los administradores principalmente; cabe destacar que el acceso a la plataforma tampoco se ha negado en ninguno de los 2 ámbitos, sólo se ha restringido las acciones que estos pueden realizar, dependiendo en qué plataforma se inicie sesión.

Por la parte de funcionalidades estas ya se habían consignado previamente en la presentación del trabajo de grado, y también se había hablado en conversaciones regulares con el administrador, por lo que formalmente el único requisito que se hizo extra, de manera formal fue la adición de la firma; vale la pena aclarar que estas sesiones esporádicas fueron de gran ayuda para la definición de las funcionalidades más importantes y que estas aunque no fueron tenidas en cuenta inicialmente se pueden contar cómo entrevistas.

Requisitos

Después del levantamiento de requisitos el cual consta de una entrevista informal con la administradora del Hogar Santa Maria, y de conversaciones regulares con esta misma se dejaron en claro los siguientes requisitos.

- Posibilidad del administrador de consultar casi cualquier información del estudiante por parte del administrador (se excluye la contraseña por motivos de seguridad).
- Todo estudiante debe contar con una firma para la correcta diligenciación de las planillas de salida y de inasistencia (una fotografía de esta).
- El estudiante será el único encargado de ingresar su información acerca de sus actividades.
- El registro en la aplicación solo será posible hacerlo un usuario con el rol de administrador.
- La gestión de los estudiantes estará a cargo solamente por los administradores.
- Contar con un inicio de sesión para los usuarios.
- Las reservas a los espacios podrán ser reservadas por cualquier usuario y estas no deben cruzarse entre sí, dentro de un rango de horas establecido (reglas internas).
- El administrador puede hacer consultas de manera general cómo:
 - Consultar a los estudiantes que están fuera de las instalaciones, que han registrado sus salidas.
 - Consultar a los estudiantes que registraron inasistencias para los horarios de alimentación.
 - Consultar las reservas futuras.
 - Las actividades de los estudiantes.

Requerimientos

Apéndice A.

Historias de Usuario, donde se desglosan los requerimientos según el tipo de usuario.

Objetivo Específico II

Este objetivo específico corresponde al diseño en el desarrollo de software.

Durante la realización de este objetivo se contó con la ayuda del asesor el cual aprobó la decisión de usar una arquitectura hexagonal, ya que el uso de interfaces para integrar las capas entre sí ayuda enormemente a la independencia de las tecnologías y la división en módulos ayuda a reducir el acoplamiento al mínimo; estas características son perfectas para que la aplicación sea consumida desde cualquier fuente, sea una aplicación web o una aplicación móvil.

Otro motivo también es la facilitación de los cambios que se puedan presentar, cómo la independencia del motor de base datos, al usar un ORM, Hibernate en este caso; también ayuda a cumplir el principio SOLID.

Apéndice B

MER de la Base de Datos, independiente del motor de base de datos.

Apéndice C

Diagrama UML del dominio, independiente de cualquier framework de programación.

La parte de este proyecto que requirió más diseño fue el Backend, el cual al ser una arquitectura hexagonal se debían seguir unas pautas para cumplir con esta arquitectura; Para la aplicación móvil al usar Angular en IONIC, la estructura básica es la por defecto, exceptuando, una carpeta que se genera en la raíz del proyecto, encargada de contener todo el código para la plataforma de Android; a nivel de estructura interna se decidió por una división entre componentes de acuerdo a sus tipos; Para la página web se decidió una división por funcionalidades, más que por componentes.

Apéndice D

Descripción de la estructura, donde se explica con más detalle cada capa de código y cada directorio, de cada repositorio.

Objetivo Específico III

La selección de las tecnologías fue un poco confusa al inicio ya que los realizadores no tenían conocimientos de la totalidad de tecnologías a usar, solo se tenía claro las necesidades por lo que fue necesario que realizaran varias consultas previas; esto sucedió más que todo con el caso de la parte móvil de la aplicación la cual tenía el requerimiento funcional de que fuera compatible con Android e IOS para garantizar una cobertura de casi el 100% de todos los habitantes del hogar.

Para tener este requerimiento en cuenta y que al mismo tiempo gastar el menor tiempo de desarrollo posible los desarrolladores escogieron IONIC, un framework que permite desarrollar aplicaciones móviles basándose en tecnologías web cómo lo son HTML, CSS y JS; este framework permite una gran independencia de los entornos de desarrollo móviles predominantes, Java/Kotlin para Android y Swift para IOS; sin omitir el hecho de que estos tampoco pueden ser dejados de lado, ya que son necesarios para la generación de las aplicaciones ya instalables y de posibles configuraciones extras.

Para la parte web la decisión fue menos confusa ya que los realizadores decidieron usar Angular, un framework con el que ya habían trabajado, y tenían conocimientos suficientes para empezar con el proyecto.

Para la parte de lógica de negocio la decisión fue entre Java o C#, pero debido a que los desarrolladores tenían más conocimientos en Java, esta fue la elección más óptima para reducir el tiempo de desarrollo gastado en curvas de aprendizaje, además de que la integración con Spring Boot lo hace especialmente potente y facilita la creación de APIs enormemente, sin dejar de lado las facilidades para usar Hibernate cómo ORM también presentes.

En cuanto a las técnicas de trabajo se optó por un modelo iterativo-incremental ya que los realizadores tuvieron una visión general del proyecto desde el inicio del desarrollo, con los componentes principales debidamente planificados, pero las funcionalidades se

fueron desarrollando de manera individual, empezando desde la más a menos importante, esto también dio pie para que las funcionalidades anteriores pudieran ser refinadas sin impactar en la aplicación de manera general; cabe destacar que los realizadores intentaron usar SCRUM inicialmente pero al compromiso de los eventos y de las entregas continuas fue impráctico al compaginarlo con las responsabilidades latentes de la universidad.

Finalmente los últimos componente por definir fueron basados en 2 características que fueran comunes, licencias de código abierto ó gratuidad, y conocimientos previos; las tecnologías escogidas fueron PostMan para la creación y prueba de las APIs; PostgreSQL cómo motor de base de datos; GIT cómo software para el control de versiones, y Junit con Mockito para la realización de las pruebas unitarias y de integración.

Objetivo Específico IV

Este objetivo específico corresponde a la programación o construcción cómo tal en las etapas del desarrollo de software.

Para la implementación del proyecto los realizadores lo dividieron en tres partes principales, la aplicación web, la aplicación móvil, y la lógica de negocio; después de que se escogieran las tecnologías y se modelará inicialmente el dominio, la primera parte que se construyó fue el dominio mismo, subsecuentemente se diseñó y construyó todas los módulos presentes en la aplicación, algunas veces en paralelo entre varias capas, otras veces en simultaneo toda una capa.

Para la la implementación de la página web esta fue iniciada casi 2 meses después de haber iniciado el desarrollo de la lógica de negocio, en los primeros momentos no fue posible enlazarlos por lo que el trabajo se enfocó en la experiencia de usuario y la interfaz gráfica inicialmente; después de que los servicios del Backend fueran consumibles se enlazaron apropiadamente y se enfocó en la creación de las respectivas funciones de la página web.

Para la aplicación de la página web fue necesario que los realizadores tomaron un tiempo previo al aprendizaje del framework de IONIC, lo cual fue más rápido de lo esperado a los conocimientos previos de estos en tecnologías web y el centrarse en aprender los conocimientos más prácticos y necesarios para la realización del proyecto; para el momento de la creación de este proyecto el Backend estaba casi terminado por lo que el enlazamiento fue inmediato y se pudo centrar el desarrollo en la interfaz de usuario y las funcionalidades.

Apéndice E

Capturas de pantalla de la aplicación móvil, describiendo cada pantalla y sus funciones dentro la aplicación.

Apéndice F

Capturas de pantalla de la aplicación web, describiendo cada pantalla y sus funciones dentro la aplicación, según sus roles.

Conclusiones

Para generar una conclusión acertada y que sea lo más objetiva posible es necesario devolverse al análisis del problema y a su planteamiento original, en este planteamiento original se planteó una plataforma web para el Hogar Juvenil Universitario Santa María con el fin de ayudar a los estudiantes y a los administradores, con el fin de agilizar ciertas tareas administrativas y de mejorar la continuidad de la comunicación entre los administradores y los estudiantes que habitan el hogar.

La idea era clara y el planteamiento fue hecho de manera apropiada, por lo que el rumbo del proyecto siempre fue claro; la decisión de las tecnologías también fue bastante ágil, para esto nos basamos en todo el conocimiento adquirido durante nuestra estancia en la universidad, por lo que todos nuestro conocimientos previos fueron de mucha ayuda durante la etapa inicial del proyecto, y durante los primeros commits de cada repositorio.

La tecnología que más demoró para haber sido escogida era la que se iba a usar en la parte móvil, ya que debía ser fácil de aprender, gratuita o por lo menos que no tuviera licencias restrictivas adquisitivamente, y que fuera híbrida, este último requerimiento era de vital importancia ya que se quería llegar a las plataformas de Android y de IOS de igual manera; pero esto conllevó a un elemento inesperado, que fue el descubrimiento de que para desarrollar aplicaciones en la plataforma de Apple es necesario tener una licencia de desarrollador, la cual tiene un costo anual de 100 dólares, algo que para un desarrollo donde la gratuidad es fundamental, es un gran golpe económicamente, por este motivo la aplicación solo es compatible con dispositivos Android.

Después de escoger las tecnologías y de haber empezado a desarrollar cada respectivo proyecto todo fue sobre ruedas, hasta que empezaron surgir nuevas dudas y de encontrarse con errores durante todo el desarrollo; Es digno de mención que la ayuda de nuestro asesor fue de vital importancia para solucionar estas dudas, y dilucidar el camino a seguir para corregir los errores que nunca dejaron de salir; solo

decir que salieron muchos errores es parcialmente falso, no solo fueron errores, también fue necesario consultar nuevas cosas en internet, revisar tutoriales para implementar nuevas funcionalidades, e inclusive comprar un curso en Udemy para aprender el framework de IONIC.

Primera conclusión: acerca de la facilidad de consultar cualquier tema que nos proporciona internet, es que no todo el contenido que aparece es útil, y que el primer resultado no siempre es el que funciona, en ocasiones fue necesario pasar un tiempo considerable probando soluciones, para ver cual era la que mejor se adaptaba o funcionaba; esto aunque gastaba tiempo valioso de desarrollo fue útil para aprender nuevos conceptos, analizar el código cómo tal y darse cuenta que muchos de los errores son autoexplicativos, y que si uno comprende bien la tecnología, puede saber cual es el problema y también una posible solución, sin necesidad inmediata de consultar en internet.

Segunda conclusión: el que los lenguajes y las tecnologías están constante desarrollo, esto es muy necesario para nosotros cómo desarrolladores, ya que las nuevas funcionalidades suelen ser muy útiles y siempre son bienvenidas, pero también quiere decir que las tecnologías no son libres de errores o de incompatibilidades, por lo que también es importante conocer estas falencias, ya que pueden provocar errores que no son solucionables o que requieren hacer parches de código que en cualquier momento pueden ser obsoletos y dejen de funcionar, causando errores, comportamientos inesperados, o inclusive que deje de funcionar.

Tercera conclusión: durante el transcurso del proyecto, y más durante las etapas finales de este proyecto se ha hecho notar la importancia de la documentación del código, y el uso de buenas prácticas de programación, esto hace que el entendimiento se agilice y que la incorporación de nuevas funciones sea más eficiente; además de la inclusión de herramientas para el versionamiento de código son excelente para la medición del progreso de la aplicación, estas mediciones son increíblemente necesarias para cuando las aplicaciones tienen cronogramas estrictos.

Cuarta conclusión: el manejo del tiempo fue una falencia de este proyecto, al no tener buenas estimaciones con su uso, no haber desarrollado un cronograma específico para el desarrollo de todo el proyecto, y tener que compaginarlo con otras actividades, generalmente académicas; lo que resultó en periodos de tiempos altamente ineficaces y eficaces, por lo tanto, la constancia el enfoque son elementos que hay que mejorar para futuros proyectos.

Quinta conclusión: tener una base sólida de conocimientos y conocer el porque y el para que de la teoría y de los conceptos permite que seamos independientes de las implementaciones; cualquier persona que sepa programar lo podrá hacer en cualquier lenguaje, después de un tiempo de acoplamiento, pero cualquier persona que sepa usar un lenguaje y copie código que funciona no significa que sea un programador capaz y diligente.

Ya para finalizar uno de los aspectos más importantes para empezar un proyecto, es que este siempre lo desafiara uno, y que nunca se está lo suficientemente preparado, cómo para completar un proyecto sin ninguna ayuda o recurso externo, pero justo después de terminarlo, los nuevos conocimientos adquiridos y la experiencia ganada se convierten en una base más amplia y firme para nuevos proyectos con nuevos retos.

Referencias bibliográficas

Johanna Pino, (03 oct,2018), *udea.edu.co*, recuperado de:
[http://www.udea.edu.co/wps/portal/udea/web/inicio/udea-noticias/udea-noticia!/ut/p/z0/fY-xDolwEIZfxYWxaUVEHYmDiXFwMAa6mAs0cgh30Bbi41t0MC4ul_v-fPcnJ7XMPsaY8A4emaANXOj0tt3t42WWqJNKk1RI6TIZb-LD6nJV8ij1fyE0YDMMOpO6ZPLm6WXes_XQjpWBSIH7pZo789nnuSD2WCK4SL2vCSuerW8MJVSmw-DXfAdrnDDOjxUCeWwDjQSC-zK8I3qwlBqeDM0WCbZoQqfsH7p4AWjLOHQ!/?](http://www.udea.edu.co/wps/portal/udea/web/inicio/udea-noticias/udea-noticia!/ut/p/z0/fY-xDolwEIZfxYWxaUVEHYmDiXFwMAa6mAs0cgh30Bbi41t0MC4ul_v-fPcnJ7XMPsaY8A4emaANXOj0tt3t42WWqJNKk1RI6TIZb-LD6nJV8ij1fyE0YDMMOpO6ZPLm6WXes_XQjpWBSIH7pZo789nnuSD2WCK4SL2vCSuerW8MJVSmw-DXfAdrnDDOjxUCeWwDjQSC-zK8I3qwlBqeDM0WCbZoQqfsH7p4AWjLOHQ!/)

Religión filipenses, *rfilipenses*, recuperado de:
<http://www.rfilipenses.com/donde-estamos/>

Videosuconianos. *Youtube*, recuperado de:
<https://www.youtube.com/watch?v=7dSiYwhsYso>

©2019 Ionic. *ionicframework*. recuperado de:
<https://ionicframework.com/framework>

Ana Zita. (2018-2019). *TodaMateria*. recuperado de:
<https://www.todamateria.com/que-es-software/>

Diego Alexander. (2017). *platz*, recuperado de:
<https://platz.com/blog/que-es-arquitectura-de-software/>

José Luján.(16 jun. 2017). *EDteam*, recuperado de:
<https://ed.team/blog/paradigmas-de-programacion>

TutorialSpoint. *TutorialSpoint.com*, recuperado de:
https://www.tutorialspoint.com/es/software_engineering/software_requirements.htm

Requeridos Blog. (20 abril, 2018). *medium.com*, recuperado de:
<https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>

Fred. (21 jul, 2017). *Velneo.es*, recuperado de:
<https://velneo.es/15-buenas-practicas-proyectos-desarrollo-software/>

Angel Robledano. (11 jun, 2019). *openwebinars.net*, recuperado de:
<https://openwebinars.net/blog/que-es-la-programacion-orientada-objetos/>

Rodrigo Paszniuk. (16 jul, 2013). *programacion.com*, recuperado de:
<https://www.programacion.com.py/varios/orientacion-a-objetos>

DigitalGuide. (02 agos, 2019). *ionos.es*, recuperado de:
<https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>

Julián Pérez Porto y Ana Gardey. (2015). *definicion.de*, recuperado de:
<https://definicion.de/plataforma-virtual/>

Alejandro Nieto. (9 feb, 2011). *xatakandroid.com*, recuperado de:
<https://www.xatakandroid.com/sistema-operativo/que-es-android>

4rSoluciones blog. (22 jun, 2013). *4rsoluciones.com*, recuperado de:
<https://www.4rsoluciones.com/blog/que-son-los-paradigmas-de-programacion-2/>

reactjs, recuperado de:
<https://es.reactjs.org/>

Miguel Angel. (29 nov, 2016). *desarrolloweb.com*, recuperado de:
<https://desarrolloweb.com/articulos/que-es-una-spa.html>

Luis del Valle Hernández. (2020). *programarfacil.com*, recuperado de:
<https://programarfacil.com/blog/que-es-un-orm/>

Colaboradores de MDN. (04 may, 2020). *MDN Web docs*, Recuperado de:
<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>

Marcos Merino. (12 jul, 2014). *ticbeat*, Recuperado de:
<https://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>

Carlos Lucena Herrera. (12 jun, 2019). *OpenWebinars*, Recuperado de:
<https://openwebinars.net/blog/que-es-el-testing-de-software/>

JWT. (01 oct, 2019). *JWT*, Recuperado de:
<https://jwt.io/introduction/>

Deyimar A. (03 sep, 2020). *HostingerTutoriales*, Recuperado de:
<https://www.hostinger.co/tutoriales/que-es-cli/>

Nicole. (2017). *platzi*, Recuperado de:
<https://platzi.com/blog/bases-de-datos-que-son-que-tipos-existen/>

Colaboradores de MDN. (07 ago, 2020). *MDN Web docs*, Recuperado de:
<https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

Bitbucket. (2020). *Bitbucket*, Recuperado de:
<https://bitbucket.org/product/es/version-control-software>

BBVA. (23 mar, 2016). *BBVA API Market*, Recuperado de:
<https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/>

Alex Walton. (09 may, 2020). *java desde cero*, Recuperado de:
[https://javadesdecero.es/avanzado/anotaciones-annotations/](https://javadesdecero.es/avanzado/ anotaciones-annotations/)

Red Hat. (2020). *Red Hat*, Recuperado de:
<https://www.redhat.com/es/topics/middleware/what-is-ide>

Hector Patricio. (23 feb, 2019). *the dojo blog*, Recuperado de:
<https://blog.thedojo.mx/2019/02/23/patrones-de-diseno-que-son-y-cuando-usarlos>

Programacion.net. (2020). *Programacion.net*, Recuperado de:
https://programacion.net/articulo/5_gestores_de_dependencias_para_el_desarrollo_web_1421

Hector Patricio. (19 mar, 2019). *the dojo blog*, Recuperado de:
<https://blog.thedojo.mx/2019/03/19/principios-de-diseno-de-software.html>

Apéndice A

Historias de Usuario para los Usuarios en general

HU-B1: Reserva de los electrodomésticos	
Como: Estudiantes o Administrador	
Quiero: Usar los espacios de manera más eficiente.	
Para: Saber quien y cuando, esta usando algun electrodomestico	
Descripción: cualquier estudiante que se esté hospedando en Santa Maria puede usar cualquier electrodoméstico, sin embargo estos requieren ser reservados como la sala de TV además de que pueden ser muy solicitados como la lavadora, por lo que sería muy conveniente poder agendarlos y saber quién los está usando.	
Prioridad: Alta	Estimación:
Criterios: Se podrá saber las reservas y quien las hizo para el uso de algunos de estos 2, también se podrán agregar nuevas personas a la cola siempre y cuando las reservas no se crucen entre sí	

HU-B2: Reserva de los espacios comunes	
Como: Estudiantes o Administrador	
Quiero: Usar los espacios de manera más eficiente.	
Para: Saber el uso de los espacios y un correcto agendamiento	
Descripción: cualquier estudiante que se esté hospedando en Santa Maria puede usar cualquier espacio o área común, sin embargo algunos de estos requieren reserva o pueden ser ocupados para eventos ajenos a los habitantes de Santa Maria, por lo que sería muy conveniente poder agendarlos y saber quién los está usando.	
Prioridad: Alta	Estimación:
Criterios: Se podrá saber todas las reservas para el uso de todos los espacios reservables, también se podrán agregar nuevas reservas siempre y cuando estas reservas no se crucen entre sí	

HU-B3: Autenticación	
Como: Administrador o Estudiante	
Quiero: Tener seguridad y acceder con mi respectiva cuenta	
Para: Garantizar la confidencialidad de los datos	
Descripción: Todas las cuentas de la aplicación ya sean administradores o estudiantes, estarán protegidas mediante un login donde sea necesario un iD de usuario y contraseña	
Prioridad: Alta	Estimación
Criterios: Todas las cuentas de los usuarios deberán tener un usuario y contraseña para su respectivo inicio de sesión; este debe garantizar que solo con las credenciales correctas se concederá el acceso, después de esto se guardará la sesión durante un tiempo determinado para evitar iniciar sesión constantemente.	

HU-B4: Cambio de contraseña	
Como: Administrador o Estudiante	
Quiero: Poder cambiar la contraseña	
Para: Garantizar la seguridad de los datos	
Descripción: Todas las cuentas de la aplicación ya sean administradores o estudiantes, estarán protegidas mediante un login donde se requerirá un usuario y contraseña y las contraseñas podrán ser cambiadas en cualquier momento	
Prioridad: Media	Estimación
Criterios: Todas las cuentas de los usuarios deberán tener un usuario y contraseña para su respectivo inicio de sesión; cuando se intente cambiar la contraseña esto solo será posible si la contraseña actual es correcta y la nueva contraseña se ha ingresado 2 veces de manera idéntica	

Historias de Usuario para los Administradores

HU-A1: Ingreso de Estudiantes	
Como: Administrador.	
Quiero: Poder Gestionar a los estudiantes residentes en Santa Maria.	
Para: Tener la información de todos los estudiantes residentes	
Descripción: cualquier estudiante que se esté hospedando en Santa Maria deberá tener toda su información albergada en la aplicación, refiérase a nombres, apellidos, horarios de clases, y demás datos personales que puedan ser requeridos.	
Prioridad: Alta	Estimación:
Criterios: El estudiante podrá editar y actualizar cualquiera de estos datos; pero solo el administrador podrá borrar y crear a los estudiantes.	

HU-A2: Horarios de los estudiantes	
Como: Administrador	
Quiero: Poder visualizar las actividades que realizan los estudiantes	
Para: Asignar las tareas y actividades a realizar de la mejor manera posible	
Descripción: cualquier estudiante que se esté hospedando en Santa Maria tiene un horario en el cual se concentran todas las clases y actividades que este debe realizar, en función de este se pueden asignar las tareas o actividades a realizar, tener una idea por adelantado de la asistencia a los eventos cuando tengan disponibilidad	
Prioridad: Media	Estimación:
Criterios: Se podrán ver los horarios de las actividades de cada estudiante, que haya registrado en sus usuarios.	

HU-A3: Gestión de labores	
Como: Administrador	
Quiero: Gestionar la realización de labores en el hogar	
Para: Poder consultar, asignar y modificar de manera rápida estas labores	
Descripción: cualquier estudiante que se esté hospedando en Santa Maria tiene la posibilidad de que se le asigne cualquier actividad de aseo, por lo que debe realizarla; estar pendiente de todas las actividades y de su correcta realización, es una tarea muy importante para los administradores que deben saber a quien le corresponde tal actividad, y así vigilar el cumplimiento de estas	
Prioridad: Media	Estimación:
Criterios: A cualquier estudiante se le podrá asignar cualquier tarea, también se podrá consultar todas las tareas y sus encargados respectivos, estas tareas también podrán ser modificables, ya sea cambiando el estudiante encargado o la tarea en sí misma.	

HU-A4: Planilla de salida	
Como: Administrador	
Quiero: Saber qué estudiantes han salido y a qué hora regresan	
Para: Saber quienes están presentes y quienes no	
Descripción: cualquier estudiante que se hospede en Santa Maria tiene situaciones en las que requiere salir de manera esporádica y/o que no está planeada; con este registro se puede saber qué estudiantes están presentes y quienes no en un momento determinado, de esta manera se puede saber quienes están presentes para las actividades que se planean y para los horarios de alimentación	
Prioridad: Alta	Estimación:
Criterios: las salidas de los estudiantes deben estar disponibles para la consulta de los Administradores, y así poder ver la hora de llegada, la hora de salida, el motivo y el lugar de la salida	

HU-A5: Consultar Inasistencias alimentación

Como: Administrador

Quiero: Poder ver las inasistencias de manera oportuna

Para: Controlar la preparación justa de los de alimentos

Descripción: Todo estudiante que haya avisado alguna futura inasistencia a alguno de los horarios de alimentación que se prestan en el hogar; se tendrá en cuenta para preparar la justa cantidad de los alimentos en cada horario respectivamente

Prioridad: Alta

Estimación

Criterios: El estudiante podrá avisar de manera previa cuando falte a alguno de los horarios de alimentación del hogar, de esta manera los administradores podrán consultar las inasistencias y ajustar la preparación de los alimentos.

Historias de Usuario para los Estudiantes

HU-E1: Crear Inasistencias alimentación	
Como: Estudiante	
Quiero: Poder informar inasistencias de manera oportuna	
Para: Avisar las inasistencias a algunos de los horarios de alimentación	
Descripción: Todo estudiante podrá avisar alguna futura inasistencia a alguno de los horarios de alimentación que se prestan en el hogar	
Prioridad: Alta	Estimación
Criterios: El estudiante podrá avisar de manera oportuna cuando falte a alguno de los horarios de alimentación del hogar; también se evitará que se hagan de manera fraudulenta al restringir estos avisos con fechas límites.	

HU-E2: Horarios de los estudiantes	
Como: Estudiante	
Quiero: Poder ingresar las actividades académicas o deportivas	
Para: Agendar las clases y actividades a realizar en la universidad	
Descripción: cualquier estudiante que se esté hospedando en Santa Maria tiene un horario en el cual se concentran todas las clases y actividades que este debe realizar, en función de este se planean sus actividades, y la asistencia a los eventos cuando tengan disponibilidad	
Prioridad: Alta	Estimación:
Criterios: Se podrán ingresar, actualizar, eliminar y ver todas las materias y/0 actividades en el horario del estudiante en cuestión	

HU-E3: Consultar actividades a realizar	
Como: Estudiante	
Quiero: Poder revisar las actividades que me fueron asignadas	
Para: Agendarme y así poder cumplirlas de eficazmente	
Descripción: Todos los estudiantes del hogar tienen la obligación de cumplir con la o las tareas que los encargados asignan, de manera eficaz y oportuna	
Prioridad: Media	Estimación
Criterios: El estudiante podrá avisar revisar las actividades que se le asignaron y ver información de la actividad como una breve descripción, el espacio en donde realizarla y una frecuencia recomendada.	

HU-E4: Consulta de los horarios de lavado de loza	
Como: Estudiante	
Quiero: Poder revisar los horarios del lavado de loza	
Para: Así saber mis respectivos turnos dado el caso	
Descripción: Todos los estudiantes del hogar son posibles candidatos a lavar la loza en cualquiera de los turnos de la loza que existen en la semana, generalmente 3 por día y 2 estudiantes por turno	
Prioridad: Alta	Estimación
Criterios: El estudiante podrá revisar los horarios de lavado de loza; en caso de alguna salida o acontecimiento imprevisto este podrá saber el nombre de otra persona y así contactarlo o hablarle para pedir un intercambio provisional	

HU-E5: Firma digital	
Como: Estudiante	
Quiero: Tener una firma digital	
Para: Firmar las planillas de manera virtual	
Descripción: Todos los estudiantes del hogar tienen la obligación de firma las planilla de salidas cuando salen del hogar en actividades que están fuera de su cronograma normal	
Prioridad: Media	Estimación
Criterios: El estudiante podrá ingresar una imagen como su firma, y cambiarla si lo cree necesario	

HU-E6: Creación de Salidas

Como: Estudiante

Quiero: Poder registrar las salidas

Para: Informar de mi salida a los administradores

Descripción: Todos los estudiantes del hogar tienen la obligación de firmar las planillas de salidas cuando salen del hogar en actividades que están fuera de su cronograma normal, indicando el motivo, lugar, fecha de salida y de llegada

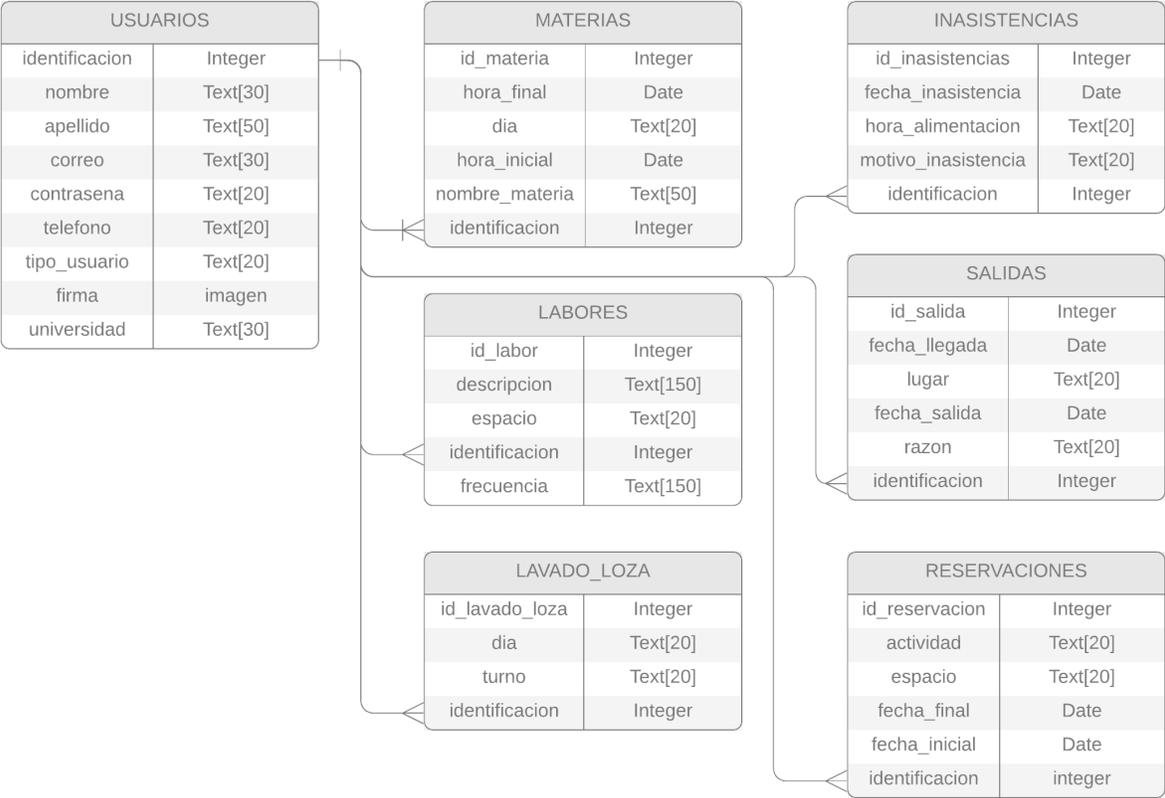
Prioridad: Alta

Estimación

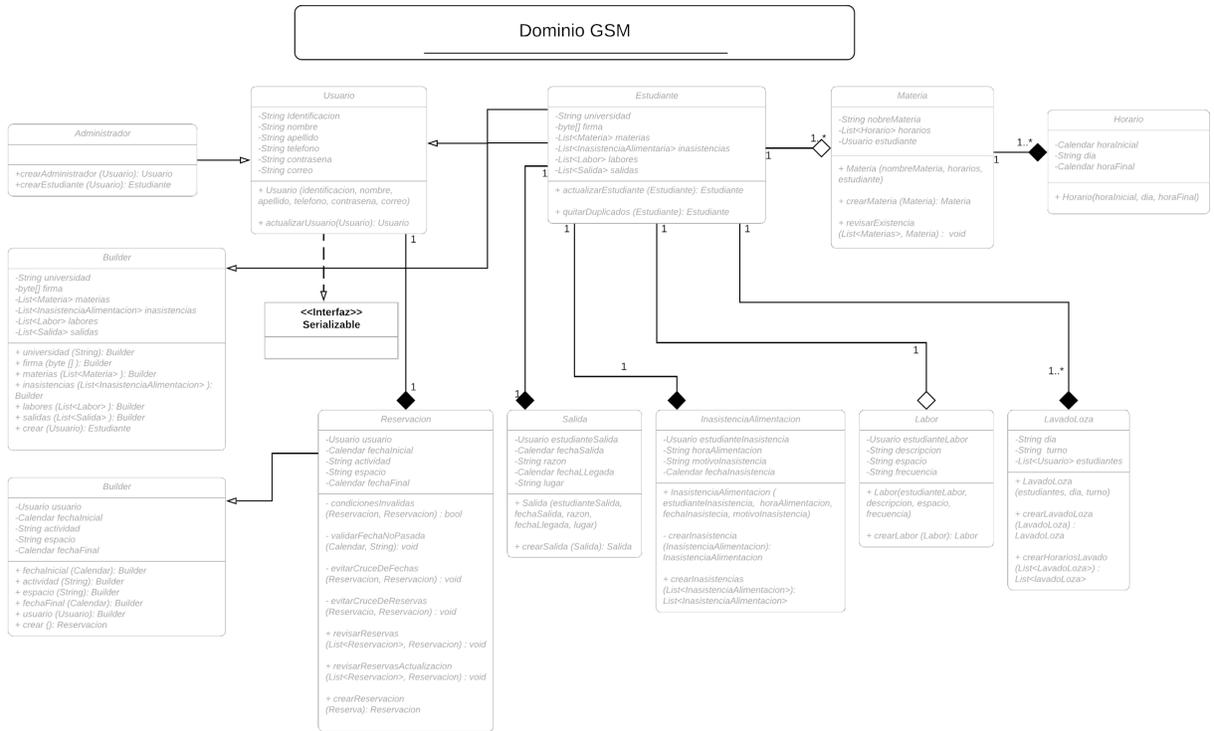
Criterios: El estudiante deberá ingresar la fecha de salida, de llegada, el motivo y el lugar para así crear una salida que el administrador podrá consultar

Apéndice B

ERD GSM



Apéndice C



Apéndice D

Estructura de la lógica de negocio

- Converter: capa encargada de desarrollar el patrón adapter entre las entidades y el dominio.
 - Interfaces: carpeta donde se agrupan las 2 interfaces que se implementan en esta capa.
- Domain: capa donde se encuentra la mayoría de la lógica del negocio y el modelamiento del sistema.
- Entity: capa donde se encuentran las entidades de la aplicación que son las encargadas de crear la persistencia de los datos de la aplicación.
- Repository: capa donde se encuentran las interfaces de las operaciones que se realizan en la base de datos, básicamente un CRUD.
- Restcontroller: capa donde se exponen los servicios REST de la aplicación para ser consumidos mediante el protocolo HTTP, en formato JSON, en esta capa también se devuelve el resultado de la operación, si es exitosa será un 200, de lo contrario será un 400 ó otro tipo de error más específicos.
- Services: capa encargada de ejecutar las llamadas de la aplicación que ingresan a través de la capa restController y devolver el resultado de la petición sea exitoso o erróneo.
 - impl: carpeta donde se agrupan las implementaciones de las interfaces.
 - interfaces: carpeta donde están las interfaces a implementar.
- Transacciones: capa mediadora entre la respuesta de la capa restController y la respuesta de los services, si es exitosa devolverá la respuesta con los datos solicitados, sin modificación alguna, de lo contrario devolverá una excepción con un mensaje especificando el error presentado, al cual el restcontroller añadirá el estado de error.
- Security: capa encargada de la seguridad de la aplicación, en esta capa se encuentra la generación y validación de los tokens JWT, la descryptación de las contraseñas de los usuarios, la configuración de los CORS y los filtros de las peticiones.

- Test: Capa donde se encuentran las pruebas unitarias y de integración realizadas.
- Transversales: estos elementos son transversales a todo el proyecto, por lo tanto no están definidos en una capa cómo tal, estos son:
 - Útil: con métodos reutilizables para el manejo de objetos, strings y fechas.
 - Exceptions: que es el manejo de Excepciones personalizadas.
 - Dto: que es donde se conservan algunos DTO que usa la aplicación en algunas de las peticiones;
 - Mensajes: donde están todos los mensajes de la aplicación centralizados.
 - Constantes: donde están algunos valores por defecto y las constantes necesarias para la autenticación del usuario.

Estructura de la aplicación móvil

- Components: En esta carpeta se ubican los componentes de Angular que son reutilizados en varias partes de la aplicación.
- Guards: Solo contiene un archivo que es un Guard, un componente propio de Angular, que permite proteger las páginas de usuarios no autorizados, impidiendo su carga, en este caso.
- MainTabs: Carpeta donde se agrupan las 4 principales pantallas de la aplicación junto, cada pantalla con una tarea en específico.
- Models: Carpeta donde estan los archivos de typescript de la aplicación que hacen parte de esta, y que pueden ser transversales, cómo lo es el caso de las constantes y de los mensajes.
- Pages: En esta carpeta están los tipo page, que son componentes propios de IONIC, estos son funciones específicas dentro de la aplicación, cómo lo puede ser el agregar la firma o el actualizar los datos personales del usuario.
- Pipes: Esta carpeta contiene varios pipes que usa la aplicación para filtrar la información de manera visual dentro de la aplicación, los pipes son componentes propios de Angular y ayudan a visualizar los datos de otras maneras, sin alteración alguna para estos.

- Services: Carpeta donde se encuentran todos los puntos de conexión entre la aplicación y el backend, los servicios en Angular permiten centralizar código para que pueda ser usado de una manera mucho más ordenada, también son muy útiles para la comunicación cliente-servidor, al tener un lugar específico donde realizar las peticiones y manejar las respuestas.
- Assets: carpeta predeterminada para guardar archivos externos cómo imágenes.
- Environments: carpeta proveída por Angular para el uso de diferentes ambientes cómo puede ser el desarrollo y el de despliegue.
- Theme: carpeta con un archivo de SCSS que contiene todos los colores usados en la aplicación.
- Resources: carpeta ubicada en la raíz del proyecto donde, se almacenan los archivos necesarios para el despliegue de la aplicación en Android o IOS; en este caso, solo Android.
- Android: carpeta ubicada en la raíz del proyecto donde se guardan los archivos generados automáticamente para Android, esta carpeta es la aplicación ya en Android, lista para ser ejecutada y desplegada.
- www: carpeta donde se guardan todos los archivos JS, que se generaron automáticamente, dejando la aplicación lista para su ejecución, en los IDE de Android o IOS.

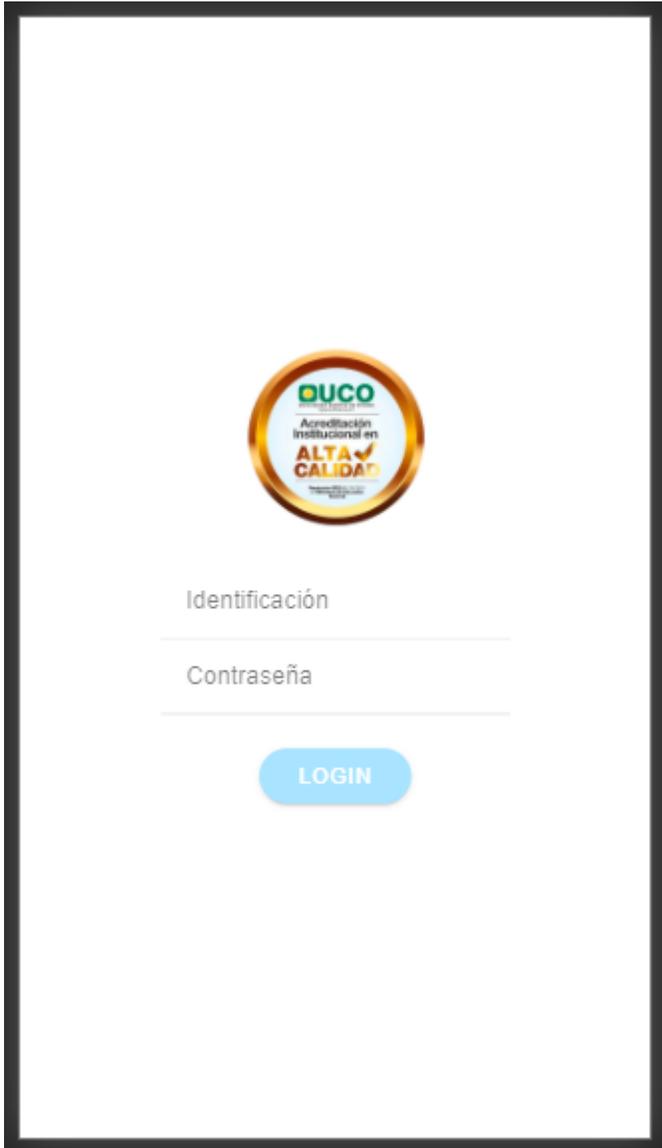
Estructura de la página web

- Component: En esta carpeta se ubican los componentes de Angular que son reutilizados en varias partes de la aplicación.
 - Core: en esta subCarpeta se encuentran los componentes iniciales, cómo el header, el menú, el footer y la barra de navegación.
 - User: Subcarpeta para agrupar las funciones o los componentes directamente relacionados con un usuario.
 - otros: los demás componentes se encuentran separados por carpetas respectivamente.
- Pipes: Esta carpeta contiene varios pipe que usa la aplicación para filtrar la información de manera visual dentro de la aplicación, los pipes son componentes

propios de Angular y ayudan a visualizar los datos de otras maneras, sin alteración alguna para estos.

- Services: Carpeta donde se encuentran todos los puntos de conexión entre la aplicación y el backend, los servicios en Angular permiten centralizar código para que pueda ser usado de una manera mucho más ordenada, también son muy útiles para la comunicación cliente-servidor, al tener un lugar específico donde realizar las peticiones y reutilizar código que será útil en diversas partes de la aplicación.
- Model: carpeta donde se encuentran todas las clases definidas en la aplicación.
- Consts: carpeta donde se centralizan partes transversales de la aplicación, cómo los mensajes y las claves que se usan para guardar datos en el localStorage.

Apéndice E.



Identificación

Contraseña

LOGIN

Login de la aplicación

Logo de la UCO, celebrando la acreditación en alta calidad

Formulario de ingreso con los campos de identificación y Contraseña

Realizadores
Andrés García - Ingeniero de sistemas UCO
Brayan Erazo - Ingeniero de sistemas UCO
Diego Alejandro - Asesor/Profesor UCO

Licencias de Código Abierto
Ionic 5 con Licencia MIT
Capacitor con Licencia MIT
Angular con Licencia MIT
Spring Boot con Licencia Apache 2.0
PostgreSQL con Licencia Open Source

Información de los realizadores

Tecnologías que fueron usadas en la aplicación móvil.

Salidas

Solo se permiten salidas entre las 04 y las 21 (en punto) :

Fecha de Salida OCT 21 18

Lugar

Fecha de Llegada OCT 21 18

Motivo
 Seleccione el Motivo

CREAR

Motivo	Fecha Salida
Lugar	Fecha Llegada
RECREATIVO	08/31
Guatapé	09/01

Salidas Inasistencias Reservas Materias

Pantalla inicial

Información que condiciona el horario de las salidas

- Campo desplegable para seleccionar la fecha de salida
- Campo para ingresar el lugar de la salida
- Campo desplegable para seleccionar la fecha de llegada
- Campo que despliega un cuadro de diálogo para seleccionar un motivo
- Botón para crear la salida con previa verificación de los datos ingresados

Lista de salida anteriores, con todas las salidas anteriores.

		CANCELAR!	ACEPTAR
AGO	19	20	00
SEP	20	21	15
OCT	21	22	30
NOV	22	23	45
DIC	23		

Formato del campo para ingresar la fecha de la salida y la de llegada

Motivo

ACADEMICO

PERSONAL

RECREATIVO

...

Cuadro de diálogo para escoger el motivo

Inasistencias

Horas minimas para avisar (en punto):
 Desayuno: 06
 Almuerzo: 11
 Cena: 18

No se aceptaran inasistencias despues de estas horas

Motivo y Hora a Faltar

Fecha Inasistencia: OCTUBRE 21

CREAR

Motivo	Hora	Fecha
PERSONAL	DESAYUNO	08/31
ACADEMICO	ALMUERZO	08/31
PERSONAL	CENA	08/31

CANCELAR! ACEPTAR

PERSONAL **DESAYUNO**

ACADEMICO ALMUERZO

RECREATIVO CENA

CANCELAR! ACEPTAR

AGOSTO	19
SEPTIEMBRE	20
OCTUBRE	21
NOVIEMBRE	22
DICIEMBRE	23

Segunda pantalla de la aplicación

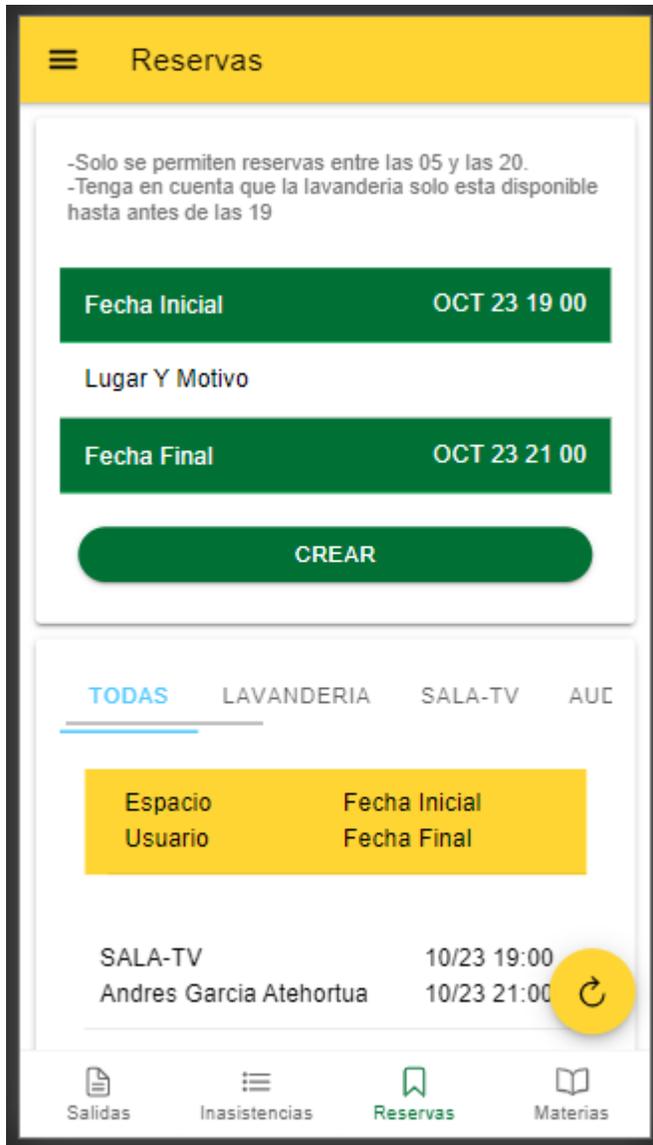
Información acerca de los intervalos de horas en los que se aceptaran las inasistencias

- Campo desplegable para ingresar el motivo y la hora de alimentación.
- Campo desplegable para ingresar la fecha de la inasistencia.
- Botón para crear la inasistencia.

Lista con inasistencias previas del estudiante

Cuadro de diálogo para seleccionar motivo y hora de inasistencia.

Cuadro de diálogo para seleccionar la fecha de la salida.



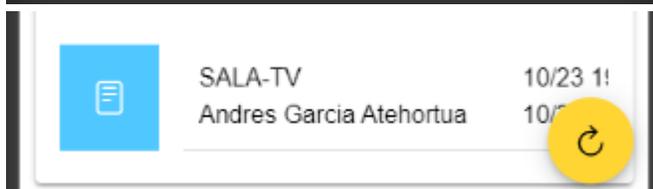
Tercera pantalla para la gestión de reservas.

Reglas para el agendamiento de las reservas.

- Campo desplegable para seleccionar la fecha inicial de la reserva
- Campo desplegable para ingresar el lugar y el motivo de la reserva.
- Campo desplegable para seleccionar la fecha final de la reserva.
- Botón para crear la reserva con previa verificación de los datos ingresados.

Lista de todas las reservas próximas, filtrables por lugar

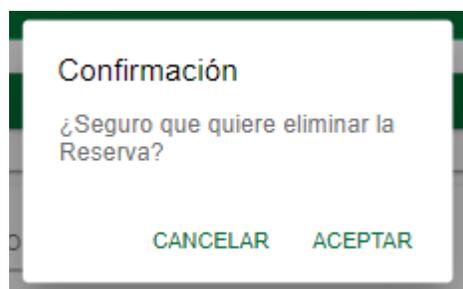
Botón para recargar las reservas.



Cada reserva del estudiante tiene un comportamiento deslizable a la izquierda para editar.



Cada reserva del estudiante tiene un comportamiento deslizable a la derecha para eliminar.



Cuadro de diálogo para confirmar el borrado de la reserva.

Actualizar Reserva

Espacio Actividad	Fecha Inicial Fecha Final
SALA-TV	10/23 19:00
PERSONAL	10/23 21:00

- Solo se permiten reservas entre las 05 y las 20.
 - Tenga en cuenta que la lavandería solo está disponible hasta antes de las 19
 - Solo se pueden cambiar las fechas

Fecha Inicial **OCT 21 22 23**

Fecha Final **OCT 21 22 23**

CANCELAR **ACTUALIZAR**

Pantalla para actualizar la reserva

Información actual de la reserva

Condiciones para actualizar la reserva

Campos de la fecha inicial que son los únicos actualizables.

Botones para regresar a la pantalla anterior o actualizar y volver a la pantalla anterior.

		CANCELAR!	ACEPTAR
AGO	19	20	00
SEP	20	21	15
OCT	21	22	30
NOV	22	23	45
DIC	23		

Campo desplegable para las fechas de la reservación, ya sea la creación o su actualización

Materias

Solo se permiten materias entre las 06 y las 20

Hora Inicial

Día
 Seleccione Un día

Hora Final

Nombre

AGREGAR HORARIO

CREAR MATERIA

Materia	Dia	Inicio	Fin
Operativos	SABADO	08	11
Dinamicos II	LUNES	12	13
Redes	MIERCOL ES	14	16
	VIERNES	17	20

Redes

MIERCOL 14
ES

VIERNES 17

MIERCOL 14 16
ES

VIERNES 17 20

Cuarta pantalla, para gestionar las reservas.

Condiciones para los horarios de las materias.

- Campo desplegable para seleccionar la hora inicial de un horario.
- Campo desplegable para seleccionar el día de un horario.
- Campo desplegable para seleccionar la hora final de un horario.
- Campo para ingresar el nombre de la materia.
- Botón para agregar un horario a la materia, se puede hacer las veces que sea necesario.
- Botón para crear la materia.

Lista de materias

Cada materia cuenta con el comportamiento de deslizar horizontalmente.

Deslizar a la izquierda para editar

Deslizar a la derecha para eliminar.

Actualizar Materia

Por Favor vuelva a ingresar los horarios nuevamente

Hora Inicial

Día
 Seleccione Un día

Hora Final

Nombre Materia
 Dinamicos II

AGREGAR HORARIO

Horarios actuales de la Materia

Dia	Inicio	Fin
LUNES	12	13

CANCELAR **ACTUALIZAR**

CANCELAR! **ACEPTAR**

15

16

17

18

19

Pantalla para actualizar la materia seleccionada.

Condiciones para editar la materia.

- Campo desplegable para seleccionar la hora inicial de un horario.
- Campo desplegable para seleccionar el día de un horario.
- Campo desplegable para seleccionar la hora final de un horario.
- Campo para ingresar el nombre de la materia, si es necesario.
- Botón para agregar un horario a la materia, se puede hacer las veces que sea necesario.

Información actual de la materia

Botones para cancelar o actualizar la materia.

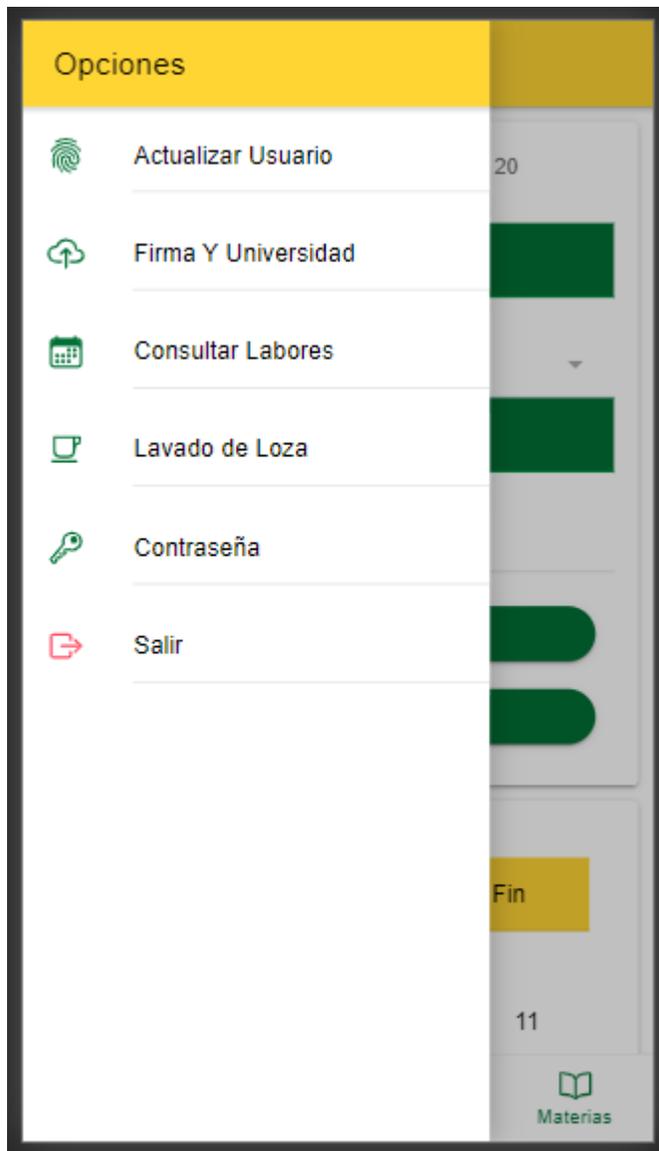
Campo diligenciable para seleccionar la hora, ya sea en la pantalla de actualizar o de crear y cualquiera de las horas.

Dialog box titled "Dia" with five radio button options: LUNES, MARTES, MIERCOLES, JUEVES, and VIERNES. At the bottom, there are two buttons: CANCEL and OK.

Cuadro de diálogo para seleccionar el día de los horarios, ya sea actualizando o creando la materia.

Confirmation dialog box titled "Confirmación" with the text "¿Seguro que quiere eliminar la Materia?". At the bottom, there are two buttons: CANCELAR and ACEPTAR.

Cuadro de diálogo para eliminar una materia.



Menú desplegable con más opciones.

← VOLVER Actualizar Datos

Ingrese solo los datos a actualizar

Identificación
1234

Nombre
Andres

Apellido
Garcia Atehortua

Telefono
3207074469

Correo
andres.garcia@uco.net.co

ACTUALIZAR

Pantalla para actualizar los datos del estudiante

Mensaje informativo.

Todos los campos son editables menos la identificación.

Botón para validar y actualizar los datos.

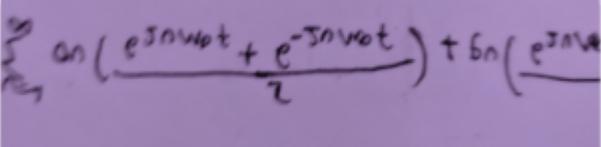
← VOLVER Firma y Universidad

No se permiten firmas de más de 1MB

Universidad
UCO

Firma

GALERIA ENVIAR



Firma Actual

Pantalla para ingresar la firma y la universidad.

Información acerca del tamaño máximo de la firma.

Campo desplegable para seleccionar la universidad.

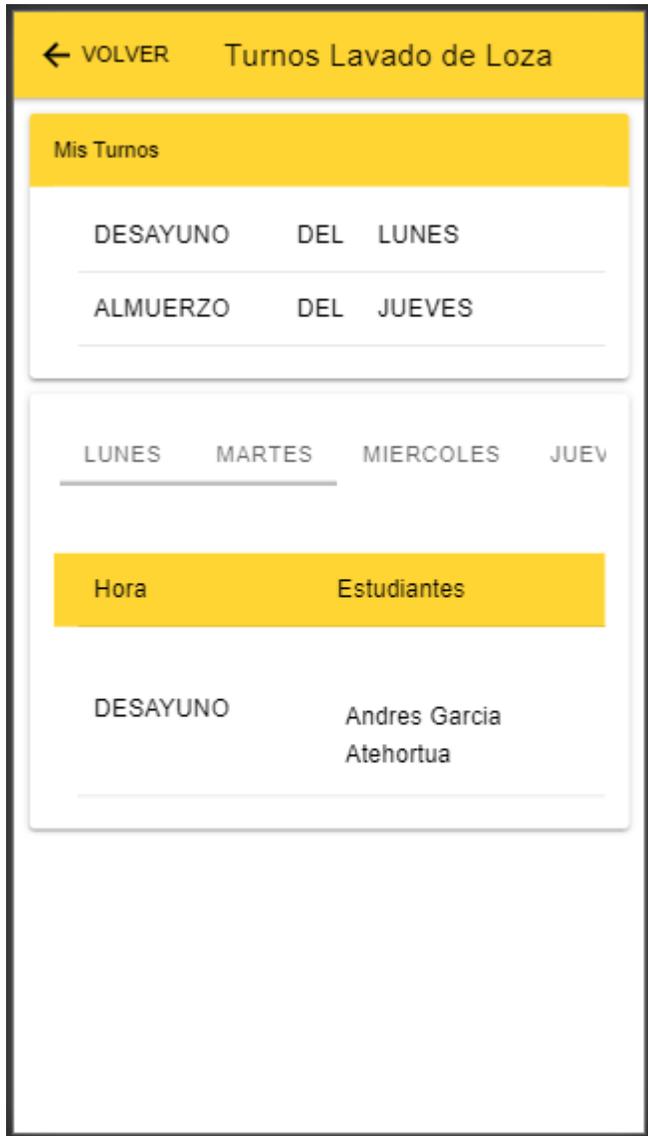
Botones para seleccionar una imagen desde la galería y enviarla.

Firma actual que el estudiante tenga ingresada.



Pantalla para visualizar las labores asignadas al usuario.

Información de la tarea a realizar.



Pantalla para revisar los horarios del

Turnos que le fueran asignados al estudiante de manera organizada.

Lista filtrable por día, de todos los turnos de lavado de loza que estén asignados, mostrando a los estudiantes responsables por cada turno.

← VOLVER Cambiar Contraseña

Contraseñas entre 6 y 30 caracteres

Identificación
1234

Contraseña Actual

Nueva contraseña

Repita la contraseña

ACTUALIZAR CONTRASEÑA

Pantalla para cambiar la contraseña

Condiciones para la contraseña.

Campos para ingresar la contraseña actual e ingresar 2 veces la nueva contraseña,

← VOLVER Cambiar Contraseña

Contraseñas entre 6 y 30 caracteres

Identificación
1234

Contraseña Actual

Nueva contraseña

Repita la contraseña

ACTUALIZAR CONTRASEÑA

Solo cuando los campos estén correctamente diligenciados se permitirá realizar el intento de cambio

Apéndice F.

Página de inicio de la página web.

Link: <https://gestionsantamaria-front.web.app/>



Login.

**HJU SANTA MARIA**LOGIN

Login

Username:

Password:

La longitud debe ser más de 6 caracteres.

Ingresar

Olvide mi contraseña

UBICACIÓN**ENCUENTRANOS EN****LICENCIAS**

Nota: Los botones del footer (pie de página) dan la ubicación del hogar, las redes sociales de ella y las licencias con la que cuenta la página web.

Olvidé mi contraseña o recuperación de contraseña.

Con el correo registrado y la identificación de tu cuenta podrás recuperar tu contraseña.



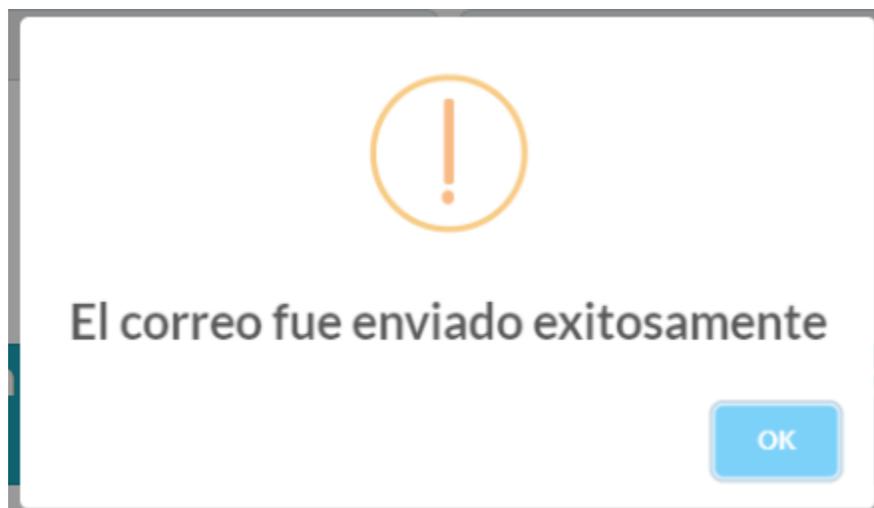
Ingresa la identificación y tu correo para recuperar tu contraseña

Correo

Identificación



Para continuar con el proceso de recuperación, se debe ingresar los datos especificados en el apartado anterior, si los datos son válidos se visualiza una alerta advirtiendo que se ha enviado un mensaje al correo como el siguiente.



Mensaje del correo.

Hola **Brayan Stiven** , tu codigo es **275819**

Después de lo anterior se desbloquean nuevos campos para continuar con la recuperación de tu cuenta, solicitando una nueva contraseña y el código que se ha enviado a tu correo.

Ingresa una nueva contraseña y el código de verificación enviado a tu correo (puede estar en Spam)

Nueva Contraseña

la longitud debe ser más de 6 caracteres.

Repetir Nueva Contraseña

la longitud debe ser más de 6 caracteres.

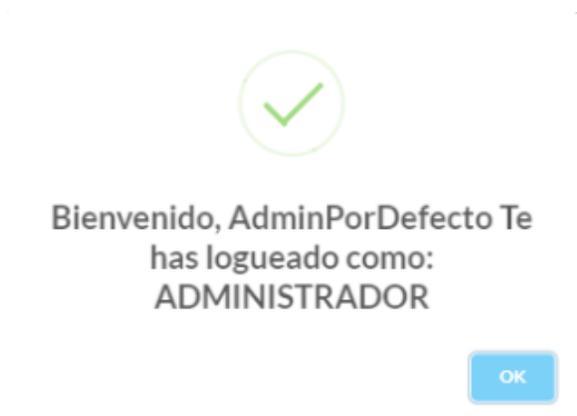
Código Recuperación

Vistas o funcionalidades desde la perspectiva de un administrador.

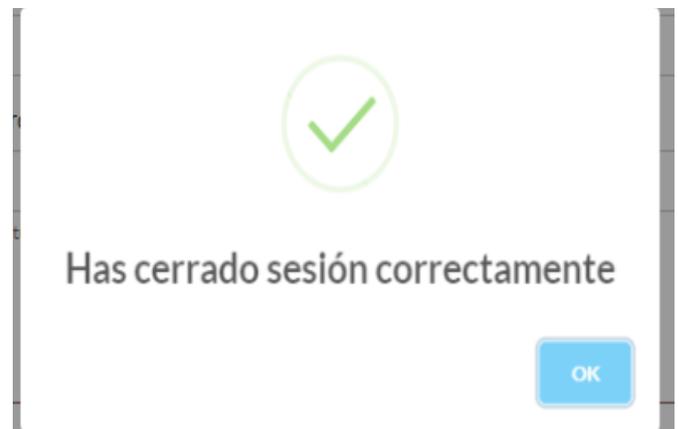
Menú con opciones o funcionalidades que puede realizar un administrador.



Alerta al loguearse como administrador.



Alerta al cerrar sesión.



Perfil de usuario.

HJU SANTA MARIA MENU PERFIL LOGOUT

Hola, Admin Por defecto !

Nombre: Admin
Apellido: Por defecto
Documento de identidad: 11223344
Correo: gestionsmuco@gmail.com
Teléfono: +57
Tipo de usuario: ADMINISTRADOR

[Cambiar contraseña](#) [Eliminar Perfil](#)

Nombres: Admin
Apellidos: Por defecto
Correo: gestionsmuco@gmail.com
Telefono: Escribe un numero telefonico
Identificacion: 11223344

[Actualizar](#)

Cuenta con una funcionalidad de actualizar datos, cambiar contraseña y eliminar la cuenta al clicar la opción correspondiente.

Cambiar contraseña.

Cambiar Contraseña

Contraseña actual: Ingrese la contraseña actual
Contraseña nueva: Ingrese la contraseña nueva
Repite la contraseña nueva: Ingrese la contraseña nueva

[Actualizar Contraseña](#)

[Cambiar contraseña](#)

Eliminar cuenta.

Eliminar Cuenta

Digita tu identificación

Ingresar tu identificación

Eliminar Perfil

Agregar un nuevo usuario: solo se puede realizar por un administrador ingresando en la opción del menú.

Nombres	Apellidos
<input type="text" value="Escribe un nombre"/>	<input type="text" value="Escribe el apellido"/>
Correo	Telefono
<input type="text" value="Escribe el correo"/>	<input type="text" value="Escribe un numero telefonico"/>
Identificacion	Tipo de usuario
<input type="text" value="Escribe la identificacion"/>	<input type="text" value="Seleccione un tipo de usuario"/>
Contraseña	<input type="checkbox"/> Acepto Terminos y Condiciones
<input type="text" value="Escribe una contraseña"/>	<input type="button" value="Terminos y Condiciones"/> <input type="button" value="Registrar"/>

Términos y condiciones, basados en el hogar Santa María.

Terminos y condiciones ×

Reglas para el ingreso a Santa María

- Prohibido el consumo de Alcohol.
- Prohibido el consumo de Alucinogenos.
- Participar en las actividades.
- Ayudar cuando sea requerido.
- Mantener buenas relaciones interpersonales

Cerrar

Gestionar usuarios: se lista los usuarios con opciones de eliminar o editar.

Lista de usuarios					
Identificacion	Nombre	Apellido	Telefono	Tipo de Usuario	Opciones
1037238115	Andres	Garcia Atehortua	3207004444	ESTUDIANTE	<button>Eliminar</button> <button>Editar</button>
987654321	Brayan Stiven	Erazo Serna	3207074444	ESTUDIANTE	<button>Eliminar</button> <button>Editar</button>

Eliminar: Se eliminará el usuario seleccionado al clickear en aceptar, para cancelar cerrar en la X.

Eliminar

×

¿ Está seguro que desea eliminar al Usuario ? Brayan Erazo identificado con: 123456?

Aceptar

Editar: Se actualizará el usuario seleccionado al clicar en aceptar, para cancelar cerrar en la X.

Editar información ✕

correspondiente al usuario Brayan Erazo identificado con 123456:

Nombre	Apellido
<input type="text" value="Brayan"/>	<input type="text" value="Erazo"/>
Telefono	Correo electronico
<input type="text" value="3115656565"/>	<input type="text" value="Brayan@gmail.com"/>

Planilla: Como administrador se puede ver la planilla de salida de los estudiantes.

Planilla de salida				
Identificacion	Fecha de salida	Fecha de llegada	Lugar	Motivo
1037238115	11/27 - 19:00	11/27 - 20:00	Rionegro	PERSONAL
1037238115	11/28 - 12:00	11/28 - 21:15	Medellin	MEDICO
1037238115	12/01 - 18:00	12/02 - 18:00	Guatapé	PERSONAL

Lavado de loza: Incluye una lista de estudiantes y el horario de lavado de loza.

Selecciona un estudiante para más opciones		
Identificacion	Nombre	Telefono
1037238115	Andres Garcia Atehortua	3207004444
987654321	Brayan Stiven Erazo Serna	3207074444

Horarios lavado de loza			
Días	DESAYUNO	ALMUERZO	CENA
LUNES	Andres Brayan Stiven		
MARTES			
MIERCOLES			
JUEVES			
VIERNES			
SABADO			
DOMINGO			

Guardar Horario

Lista de estudiantes: click al estudiante para más opciones.

Selecciona un estudiante para más opciones		
Identificacion	Nombre	Telefono
1037238115	Andres Garcia Atehortua	3207004444
987654321	Brayan Stiven Erazo Serna	3207074444

Opciones.

Que deseas hacer con Brayan Stiven Erazo Serna identificado con: 987654321

Asignar lavado loza Ver horario materias

Ver horario materias: Se visualiza el horario de materias correspondiente al estudiante seleccionado como se muestra a continuación.



Materias del estudiante Camilo Andres Erazo identificado con: 12345

Horarios materias

Nombre materia	Dia	Hora inicial	Hora Final
Calculo	MIERCOLES	23	01

Camilo Andres

Asignar al lavado de loza: Opción para elegir un turno y un día correspondiente al horario de lavado de loza.



Asignar a Brayan Stiven Erazo Serna identificado con: 987654321?

Seleccione un día

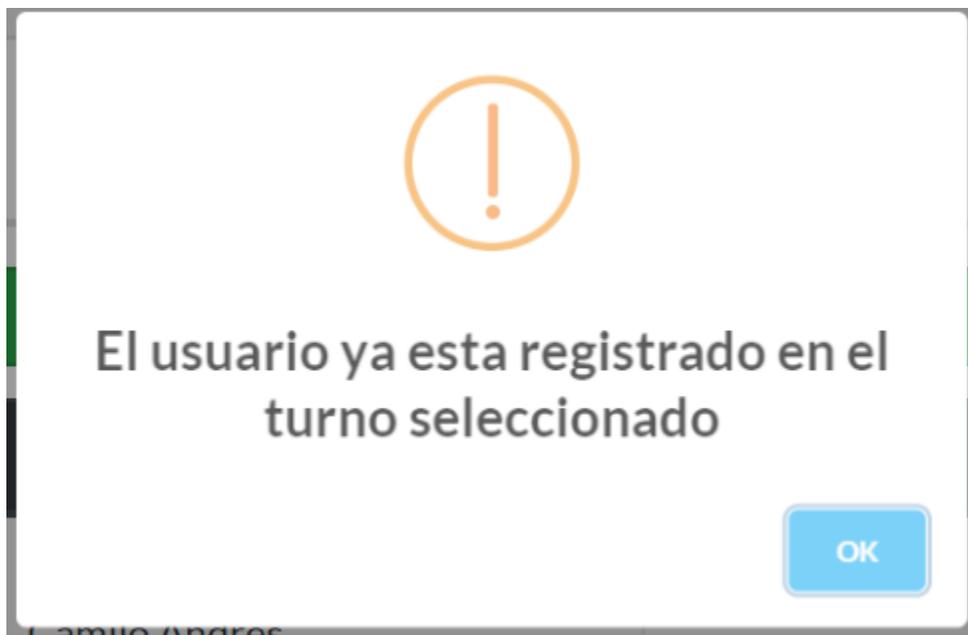
Seleccione un turno

LUNES

ALMUERZO

Asignar

Al clickear se asigna el estudiante al turno correspondiente (sólo en el caso en que no esté en el turno que se desea agregar, en caso contrario se muestra una alerta de que el estudiante ya se encuentra en dicho turno, como se muestra a continuación).



Ver horarios de materias.

Se visualiza el horario de materias del estudiante seleccionado.

Materias del estudiante Andres Garcia Atehortua identificado con: 1037238115

Horarios materias

Nombre materia	Día	Hora inicial	Hora Final
Dinamicos II	LUNES	16	18
Etica profesional	JUEVES	18	20

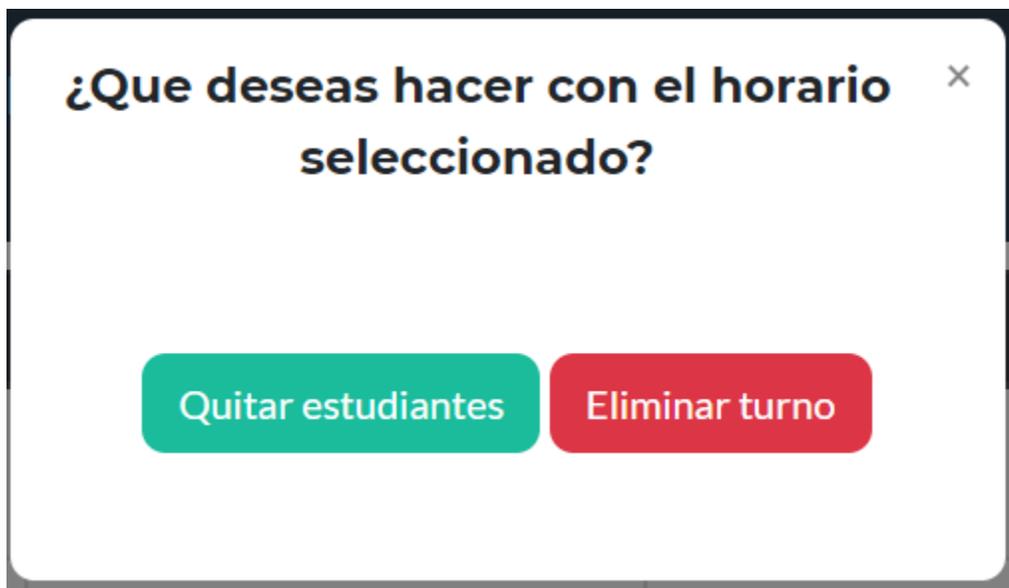
Horarios lavado de loza:

se visualiza el horario de lavado de loza registrado, se desea asignar más estudiantes ir a la opción anterior de asignar estudiantes y si desea guardar el horario clicar en Guardar Horario.

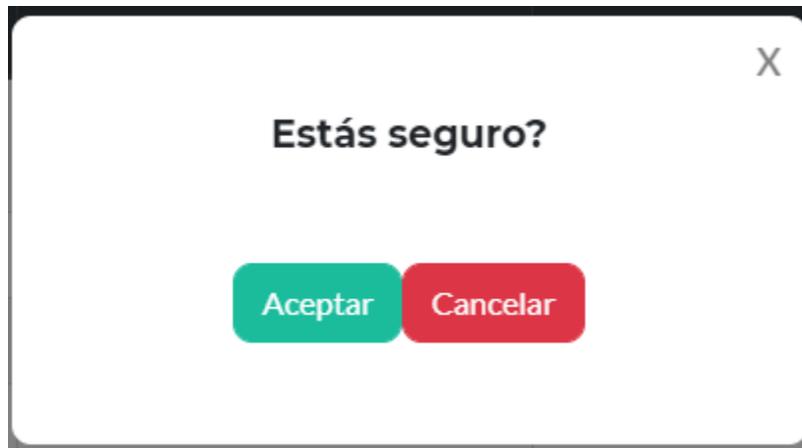
Horarios lavado de loza			
Días	DESAYUNO	ALMUERZO	CENA
LUNES	Brayan Camilo Andres		
MARTES	Brayan		
MIERCOLES			
JUEVES			
VIERNES			
SABADO			
DOMINGO			

Guardar Horario

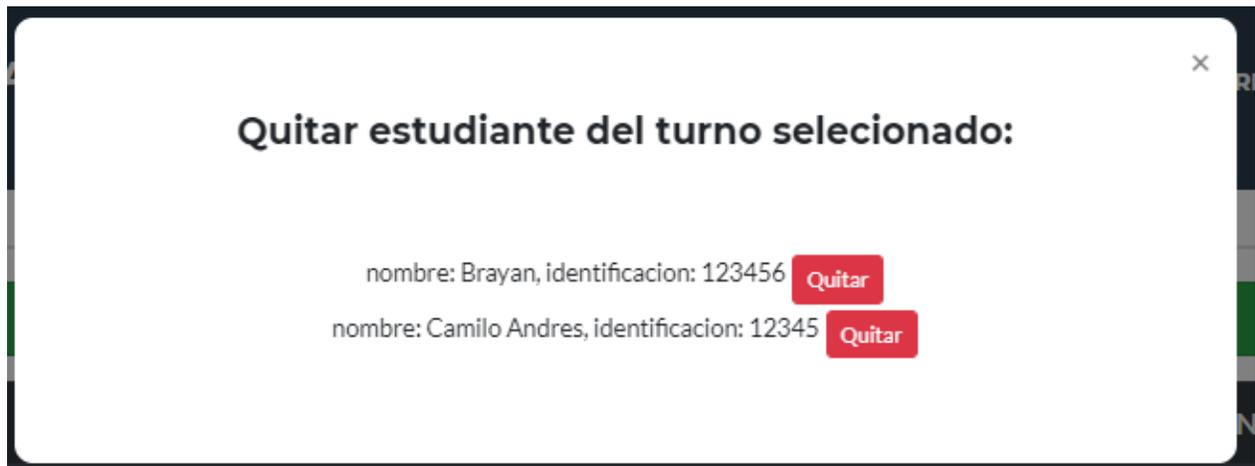
En el apartado anterior al clicar en una celda de un turno se puede observar las siguientes opciones.



Eliminar Turno: Elimina el turno seleccionado con un mensaje de seguridad para verificar si se está seguro de la opción seleccionada, al dar en aceptar se elimina el turno.

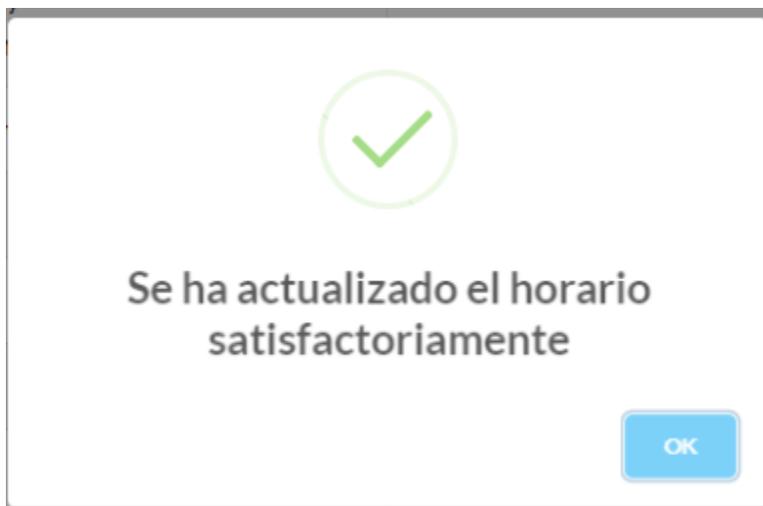


Quitar estudiantes: Se visualizan los estudiantes en dicho turno con un botón de quitar un estudiante de dicho turno (la opción anterior eliminar turno, elimina todos los estudiantes del turno seleccionado) como se muestra a continuación.

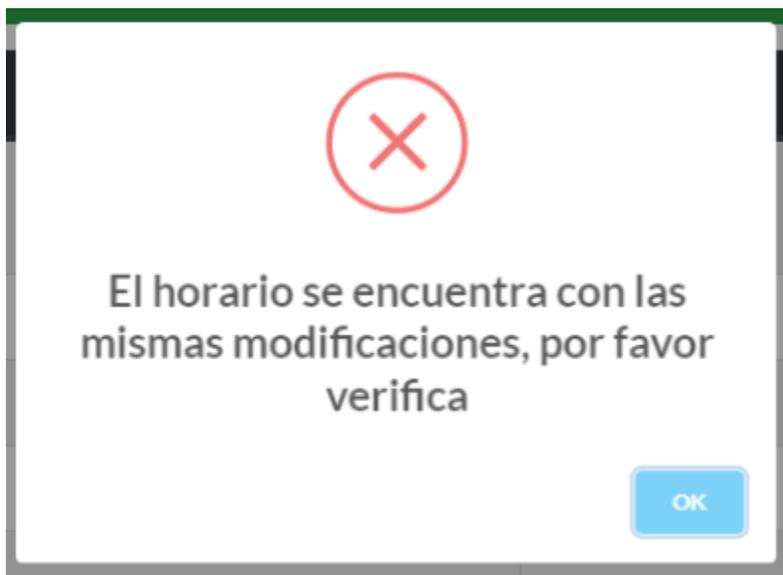


Botón guardar horario: El botón de guardar horario en el apartado de horarios lavado de loza, al ser clickeado y todo está correcto o no se encuentra ninguna modificación se visualizará una alerta respectivamente como las siguientes.

Horario guardado correctamente.



En caso de que el horario no cuente con ninguna modificación.



Opción reservas lavadora: Se visualiza las reservas de la lavadora y el botón de registrar reserva.

Reservas de la lavadora

Reservas realizadas

nombre	Fecha de inicial	Fecha de final	Lugar	Motivo
Brayan Stiven	11/18 - 15:06	11/18 - 17:06	LAVANDERIA	PERSONAL

Registrar nueva reserva

Botón registrar nueva reserva: Se abre un modal o pequeña ventana para suministrar una fecha de inicio y final estimada de la reserva como la que se muestra a continuación.

Crear Reserva Brayan Stiven

Fecha de inicio: '11/2020 01:55 p.m.

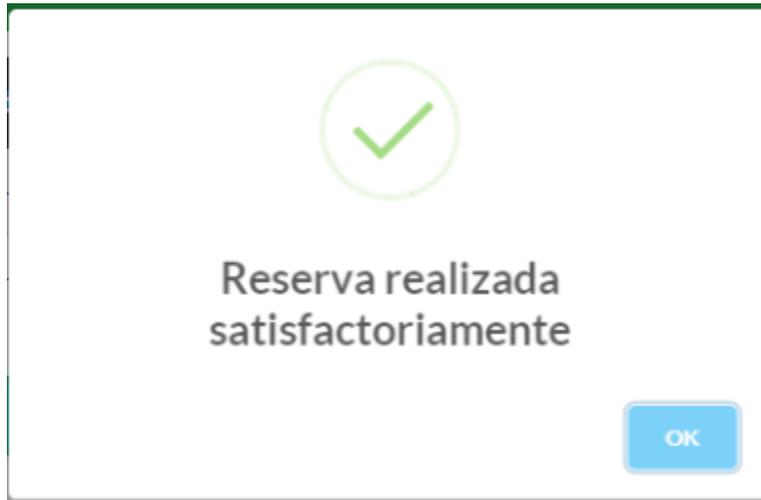
Fecha de final: '11/2020 12:55 p.m.

Motivo o razón: PERSONAL

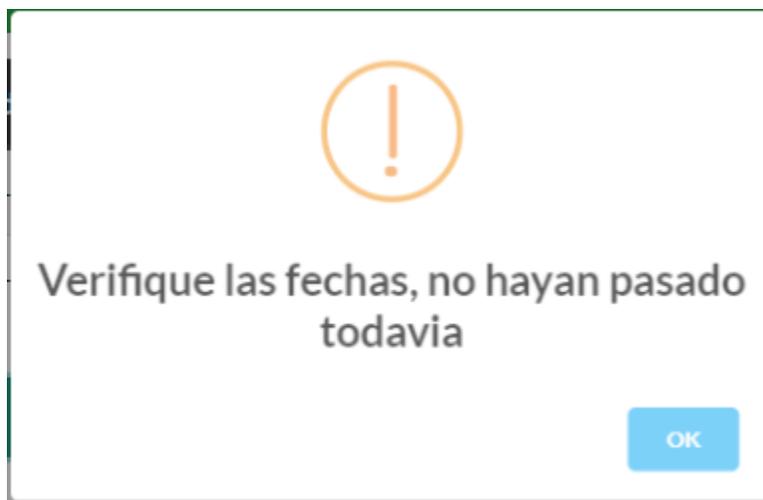
Lugar: LAVANDERIA

Reservar

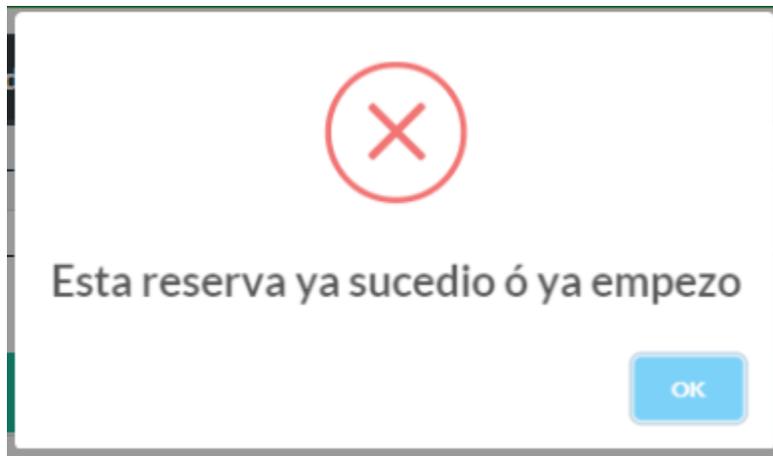
Botón reservar: Si las fechas se encuentran correctamente se visualiza la alerta o mensaje como la siguiente.



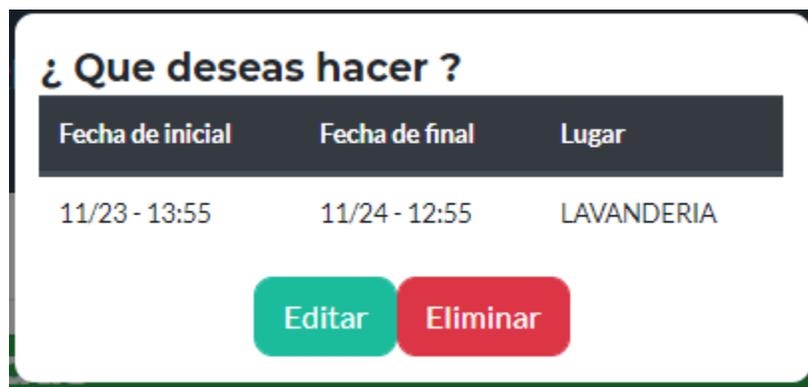
en caso contrario una alerta como la siguiente.



En caso de haber registrado una reserva y ya sucedió o empezó se visualizará una alerta, pero si aún no ha pasado de la fecha registrada se puede editar o eliminar la reserva, al realizar un click la reserva, como se muestra a continuación respectivamente.



Si desea editar o eliminar una reserva de la lavandería, selecciónala y si la reserva no ha sucedido o no ha empezado se visualizará una ventana o modal como el siguiente.



Al editar se visualizará la siguiente ventana que se comporta algo similar como el botón de registrar nueva reserva.



Solo se pueden editar las fechas ×

Fecha de inicial	Fecha de final
11/24 - 17:54	11/25 - 17:54

Fecha inicial Actual: 

Fecha Final Actual: 

Motivo o razón:

Lugar:

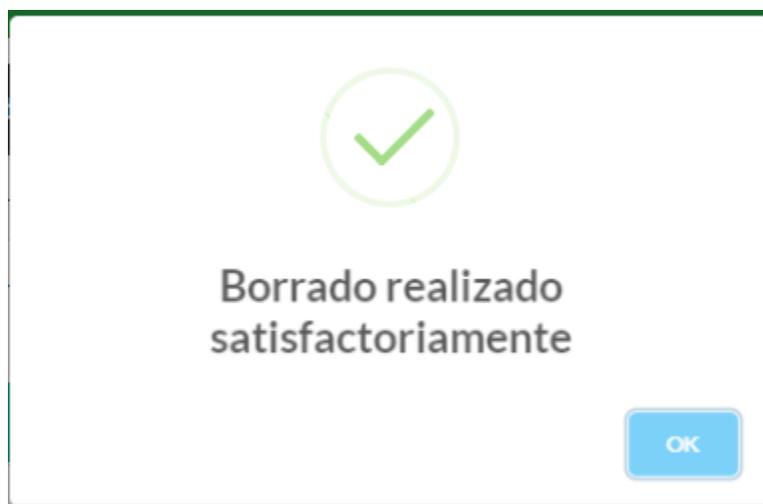
En caso de clickear eliminar, la reserva se visualiza en la siguiente ventana como seguridad para proceder con la eliminación.



¿ Desea Borrar esta reserva ? Bryan Stiven Erazo Serna ×

Fecha de inicial	Fecha de final	Lugar	Motivo
11/24 - 17:54	11/25 - 17:54	LAVANDERIA	PERSONAL

en caso de cancelar cerrar el modal y al proceder con la eliminación se muestra la siguiente alerta.



Reservas de espacios: Opción que permite visualizar y realizar reservas de los espacios o lugares en común del hogar, ya sea de entretenimiento, personal o académico.

**HJU SANTA MARIA**MENU PERFIL LOGOUT

Reservas de espacios

Reservas realizadas

nombre	Fecha de inicial	Fecha de final	Lugar	Motivo
Camilo Andres	10/26 - 18:00	10/26 - 20:00	SALA-TV	RECREATIVO
Brayan Stiven	11/04 - 14:00	11/04 - 16:00	AUDITORIO	PERSONAL

Registrar nueva reserva

Botón nueva reserva: Permite crear una nueva reserva a los espacios en común del hogar (opción semejante a las reservas de la lavadora).

Crear Reserva Brayan Stiven

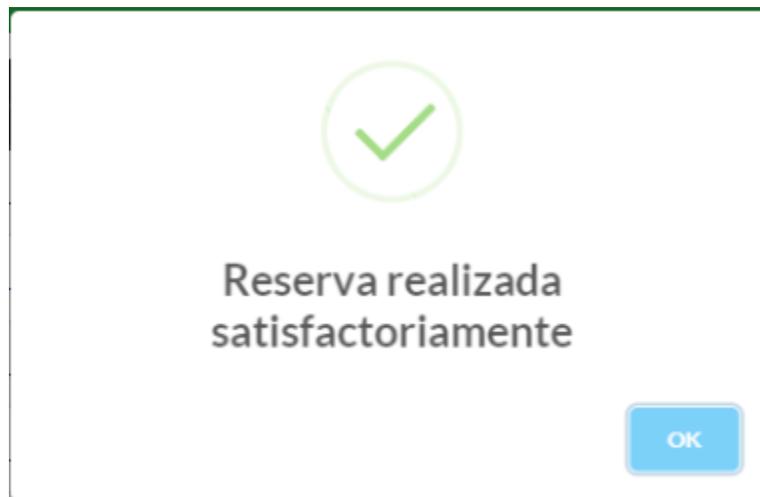
Fecha de inicio: dd/mm/aaaa --:-- ----

Fecha de final: dd/mm/aaaa --:-- ----

Motivo o razón: PERSONAL

Lugar: SALA-TV, SALA-INFORMATICA, AUDITORIO, SALON-AMARILLO, SALON#4, SALON#3

Al ajustar los campos correctamente se habilita el botón de reserva y poder realizar la reserva.



Al seleccionar una reserva que no ha sucedido o comenzado se puede editar o eliminar.

¿ Que deseas hacer ?

Fecha de inicial	Fecha de final	Lugar
11/24 - 18:38	11/24 - 22:38	SALA-INFORMATICA

[Editar](#) [Eliminar](#)

Función editar: permite editar la reserva seleccionada, el botón de editar reserva se habilita al configurar los campos de fecha inicial y final.

Solo se pueden editar las fechas ×

Fecha de inicial	Fecha de final
11/24 - 18:38	11/24 - 22:38

Fecha inicial Actual: 

Fecha Final Actual: 

Motivo o razón

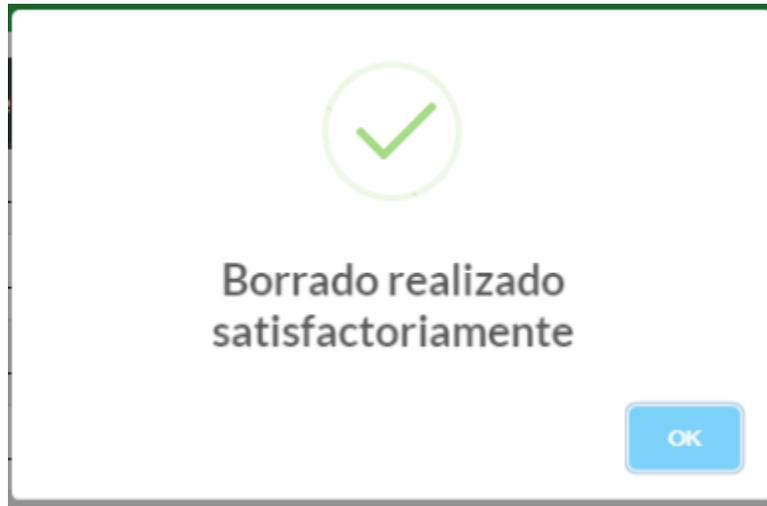
Lugar

[Editar Reserva](#)

Función eliminar reserva: Al clicar en eliminar de la ventana o modal **¿Que desea hacer?**, se visualiza un nuevo modal de seguridad para eliminar la reserva.



Si se cliqua en eliminar surgirá una alerta de la siguiente manera (Para cancelar cerrar el modal de seguridad).



Labores: Se visualiza el listado de estudiantes y las labores asignadas.

al seleccionar un estudiante o una labor se abre un modal o ventana con nuevas opciones.

**HJU SANTA MARIA**MENU PERFIL LOGOUT

Estudiantes, seleccione un estudiante para más opciones

Identificación	Nombre Completo	Telefono
1037238115	Andres Garcia Atehortua	3207004444
987654321	Brayan Stiven Erazo Serna	3207074444

Labores asignadas

Identificación	Nombre	Espacio	Descripcion	Frecuencia
1037238115	Andres Garcia Atehortua	SALA-TV	Hacer Aseo general, organizar los muebles, barrer y trapear, recoger la basura	2 ó 3 veces, por semana

UBICACIÓN


ENCUENTRANOS EN
    

LICENCIAS


Opciones al seleccionar a un estudiante.

¿ Que desea hacer con ? ×

**Brayan Stiven Erazo Serna con
identificación 987654321**

Asignar laborVer horario materias

Asignar labor:

Asignar al estudiante Brayan Erazo identificado con: 123456?

Espacio:
SALA-TV

Descripcion:
Agregue una descripcion de la labor

Frecuencia:
Agregue una Frecuencia de la labor, ej: 2 días por semana

Asignar

Ver horario materias:

Materias del estudiante Camilo Andres Erazo con identificación 12345

Horarios materias

Nombre materia	Dia	Hora inicial	Hora Final
Calculo	MIERCOLES	23	01

Opciones de labores asignadas.

¿Que deseas hacer con la labor de?
Brayan Erazo

Editar labor Eliminar Labor Actual

Editar labor:

Editar la labor del estudiante Brayan Erazo identificado con: 123456

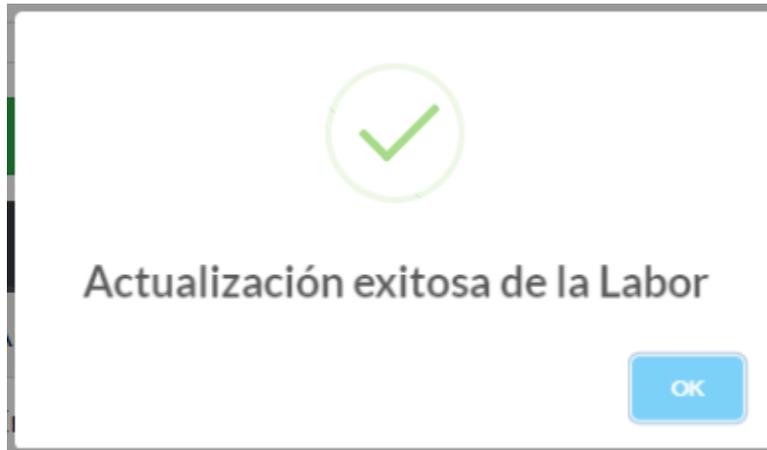
Espacio:
SALA-TV

Descripcion:
Agregue una descripcion de la labor

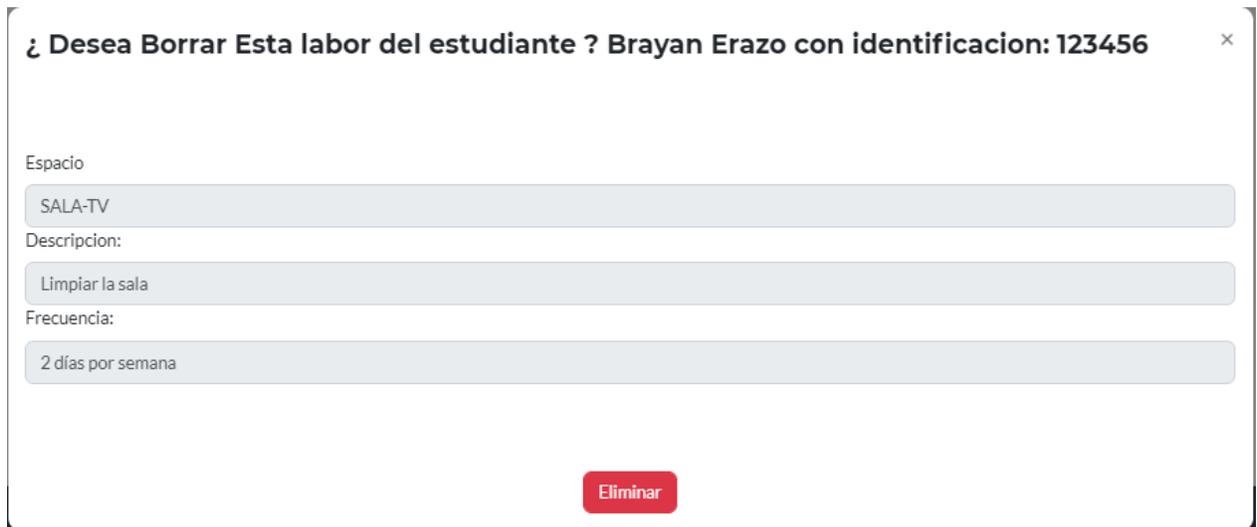
Frecuencia:
Agregue una Frecuencia de la labor, ej: 2 días por semana

Editar

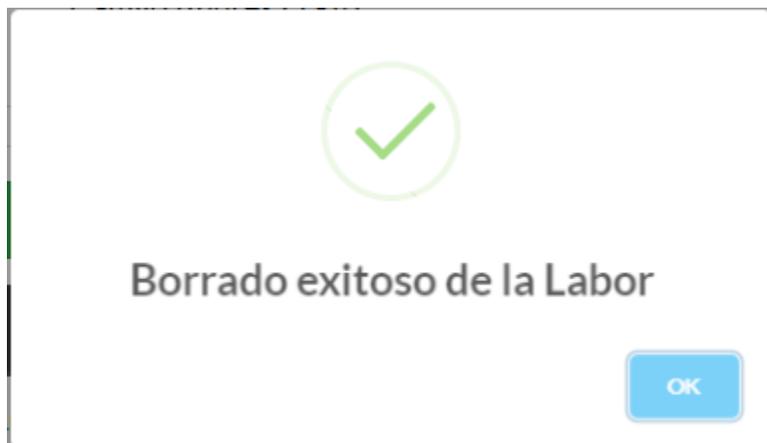
Al actualizar la labor correctamente.



Eliminar labor actual:



Al borrar exitosamente.

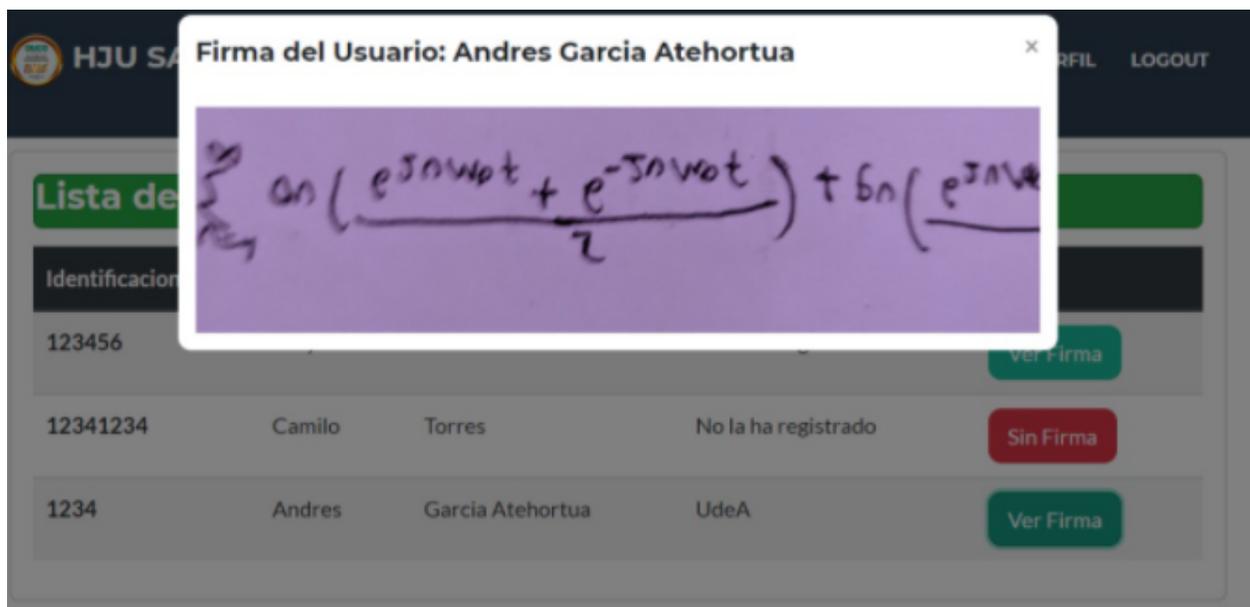


Consultar firmas: se lista a los estudiantes con su respectivos datos y la opción de visualizar la firma.



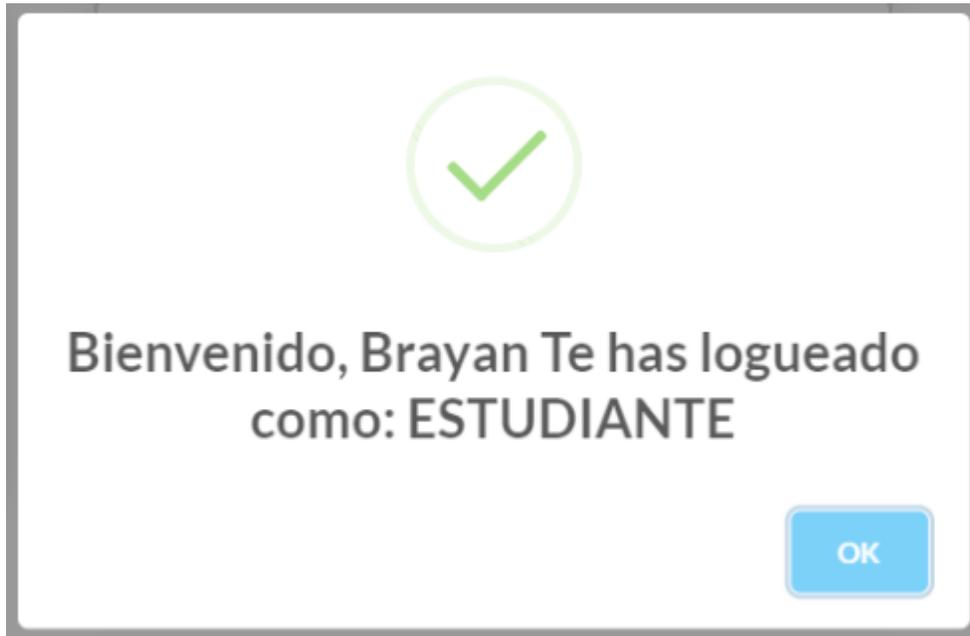
Identificación	Nombre	Apellido	Universidad	Firma
123456	Brayan	Erazo	No la ha registrado	Ver Firma
12341234	Camilo	Torres	No la ha registrado	Sin Firma
1234	Andres	Garcia Atehortua	UdeA	Ver Firma

Visualizar firma: Se visualiza la firma del estudiante seleccionado. (imagen de ejemplo de cómo se vé la firma).



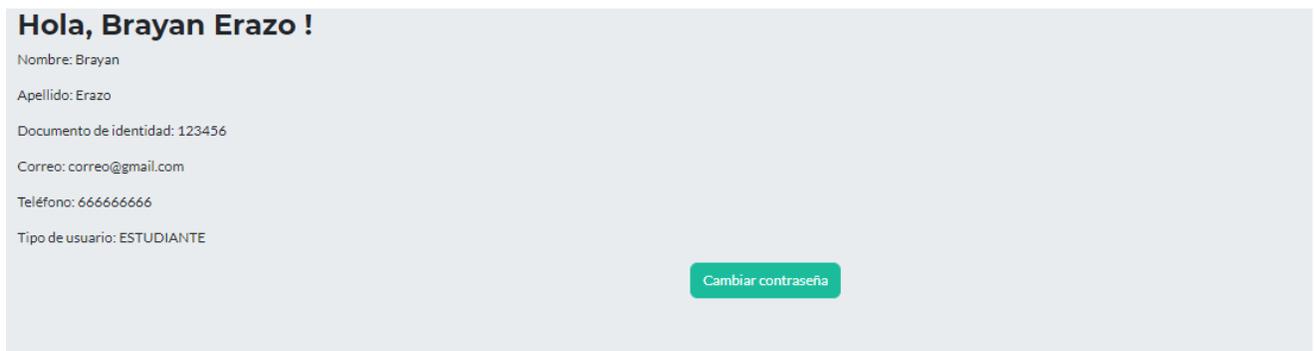
Vistas o funcionalidades desde la perspectiva de un estudiante.

Al loguearse como estudiante se visualiza el siguiente mensaje.



Perfil: Se visualiza la información del estudiante, con la opción de cambiar la contraseña, visualizar firma y actualizar algunos datos.

Si el estudiante no cuenta con una firma se visualiza el perfil de la siguiente manera.



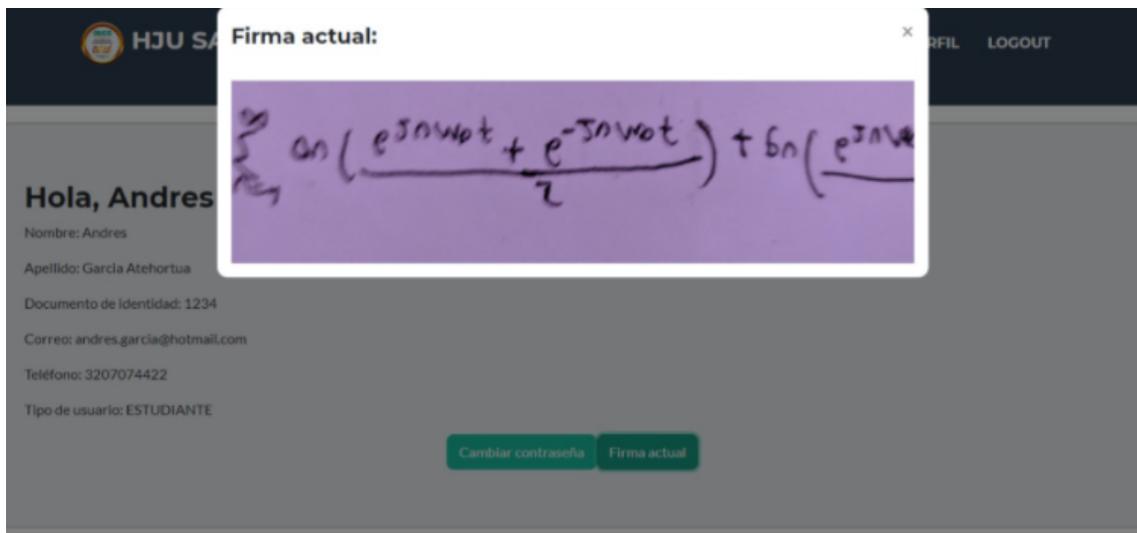
Nombres	<input type="text" value="Brayan"/>	Apellidos	<input type="text" value="Erazo"/>
Correo	<input type="text" value="correo@gmail.com"/>	Telefono	<input type="text" value="666666666"/>
Identificacion	<input type="text" value="123456"/>	<input type="button" value="Actualizar"/>	

Si el estudiante cuenta con una firma se visualiza el perfil de la siguiente manera.



The screenshot shows the user profile page for HJU Santa Maria. The header includes the university logo and name, and navigation links for MENU, PERFIL, and LOGOUT. The main content area displays the user's name, "Hola, Andres Garcia Atehortua !", followed by their personal details: Nombre: Andres, Apellido: Garcia Atehortua, Documento de Identidad: 1234, Correo: andres.garcia@hotmail.com, Teléfono: 3207074422, and Tipo de usuario: ESTUDIANTE. At the bottom, there are two buttons: "Cambiar contraseña" and "Firma actual".

Firma actual:



This screenshot shows the same user profile page as above, but with a modal window open for updating the signature. The modal is titled "Firma actual:" and contains a handwritten mathematical formula on a purple background. The formula is
$$a_n \left(\frac{e^{jn\omega t} + e^{-jn\omega t}}{2} \right) + b_n \left(\frac{e^{jn\omega t}}{2} \right)$$
. The background profile information is dimmed.

Consultar inasistencia.

el administrador puede observar las personas que van a estar ausentes un día determinado y en un horario destinado para comer.

**HJU SANTA MARIA**MENU PERFIL LOGOUT

Inasistencias realizadas

Nombre	Fecha Inasistencia	Hora Inasistencia	Motivo
Andres	2020/11/27	ALMUERZO	PERSONAL
Andres	2020/11/28	DESAYUNO	PERSONAL
Andres	2020/12/01	ALMUERZO	ACADEMICO
Andres	2021/01/20	CENA	ACADEMICO

UBICACIÓN **ENCUENTRANOS EN**     **LICENCIAS** 

Cambiar contraseña:

Cambiar Contraseña ×

Contraseña actual Contraseña nueva

Ingrese la contraseña actual Ingrese la contraseña nueva

Repite la contraseña nueva

Ingrese la contraseña nueva

Actualizar Contraseña

Menú estudiantes: el estudiante tiene funciones restringidas en comparación con el administrador.

 HJU SANTA MARIA MENU PERFIL LOGOUT



Planilla De Salida
Ingreso de Salidas



Lavado de loza
Horarios de Loza asignados a cada Estudiante



Uso de la lavadora
Consulta de reservas proximas y creación de reservas para el uso de la lavadora



Gestionar salas
Consulta de reservas proximas y creación de reservas para el uso de los espacios del hogar



Gestionar labores
Consulta de las labores asignadas como estudiante



Gestionar materias
Consulta de mis materias



Gestionar Inasistencias
Creación de Inasistencias

Planilla de salida: Se visualiza las salidas realizadas por el estudiante y adicionalmente un botón de registrar nueva salida.

Planilla de salida				
Identificacion	Fecha de salida	Fecha de llegada	Lugar	Motivo
123456	11/24 - 02:00	11/24 - 05:00	Rionegro	ACADEMICO

Registrar nueva salida

Botón registrar nueva salida: abre el siguiente modal, que se debe completar para que el botón de Registrar salida se habilite y continuar con el proceso de registro.

Ingrese los todos los datos para crear una nueva salida Brayan

Fecha de salida: '11/2020 12:59 a. m. 

Fecha de llegada: '11/2020 12:59 a. m. 

Motivo o razón: PERSONAL 

Lugar: Rionegro

Registrar salida

Al registrar se informa con una alerta el éxito de la operación.



Registro de la salida realizada satisfactoriamente



Horario de lavado de loza: El estudiante solo puede visualizar el horario que han realizado los administradores.

Horarios lavado de loza			
Días	DESAYUNO	ALMUERZO	CENA
LUNES	Brayan Camilo Andres		
MARTES	Brayan		
MIERCOLES			Camilo Andres
JUEVES			
VIERNES			
SABADO			
DOMINGO			

Reservas de la lavadora: el estudiante puede registrar o visualizar las reservas que se han realizado para el uso de la lavadora.

Reservas de la lavadora

Reservas realizadas

nombre	Fecha de inicial	Fecha de final	Lugar	Motivo
Brayan	11/25 - 01:08	11/25 - 04:08	LAVANDERIA	PERSONAL

Registrar nueva reserva

Botón registrar nueva reserva: Abre un modal para proceder con la creación de una reserva (Al completar los datos solicitados se habilita el botón Reservar)

Crear Reserva Brayan

Fecha de inicio: 11/2020 01:10 a. m.

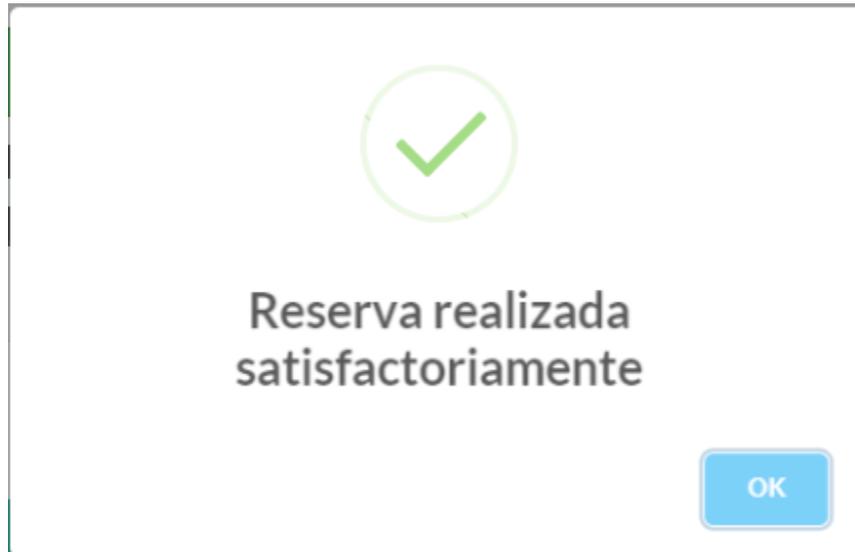
Fecha de final: 11/2020 04:11 a. m.

Motivo o razón: PERSONAL

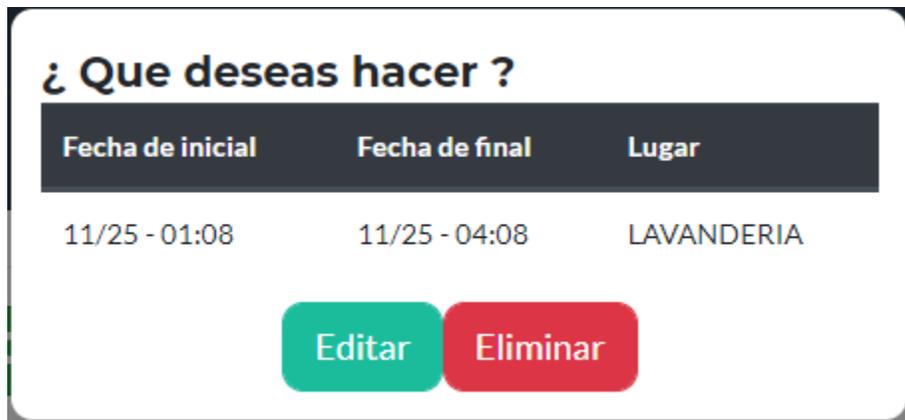
Lugar: LAVANDERIA

Reservar

Al clicar en Reservar se muestra un mensaje de éxito como se muestra a continuación.



Las reservas pueden ser editadas o eliminadas por el estudiante que las creó, siempre y cuando no haya comenzado o haya terminado.



Editar: Solo se permite editar las fechas.

Solo se pueden editar las fechas ×

Fecha de inicial	Fecha de final
11/26 - 01:10	11/26 - 04:11

Fecha inicial Actual: 

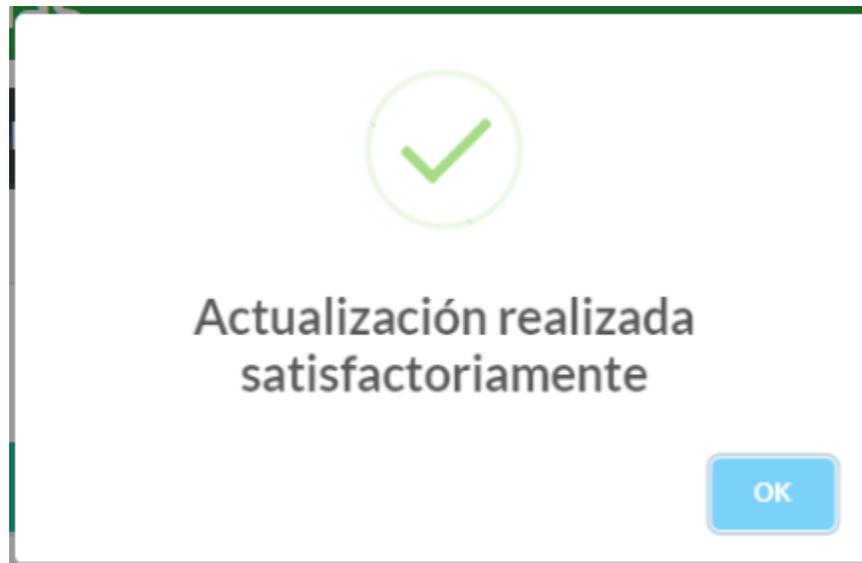
Fecha Final Actual: 

Motivo o razón:

Lugar:

Editar Reserva

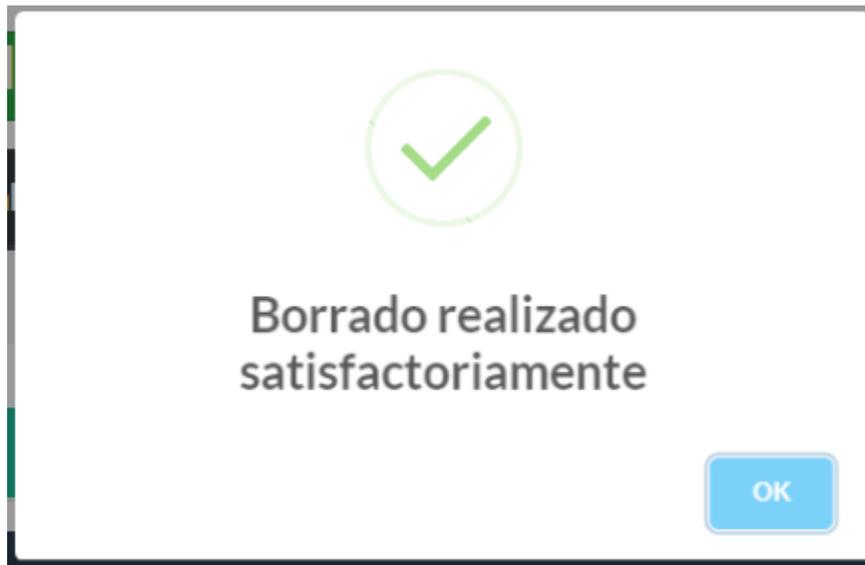
Actualización exitosa.



Eliminar: Se visualiza un modal de confirmación para borrar la reserva realizada.



Borrado exitoso.



Reservas de espacios: Se visualiza las reservas realizadas y solo el estudiante logueado puede modificar las reservas que ha hecho y siempre y cuando cumpla con que las fechas no hayan pasado o iniciado.

Reservas de espacios

Reservas realizadas

nombre	Fecha de inicial	Fecha de final	Lugar	Motivo
Brayan Stiven	11/24 - 19:50	11/24 - 22:50	SALA-INFORMATICA	PERSONAL
Brayan Stiven	11/25 - 19:51	11/26 - 19:51	SALA-TV	PERSONAL
Brayan Stiven	11/24 - 19:52	11/25 - 19:52	AUDITORIO	ACADEMICO

[Registrar nueva reserva](#)

Botón registrar nueva reserva: Abre un modal para proceder con la creación de una reserva (Al completar los datos solicitados se habilita el botón Reservar)

Crear Reserva Brayan

×

Fecha de inicio

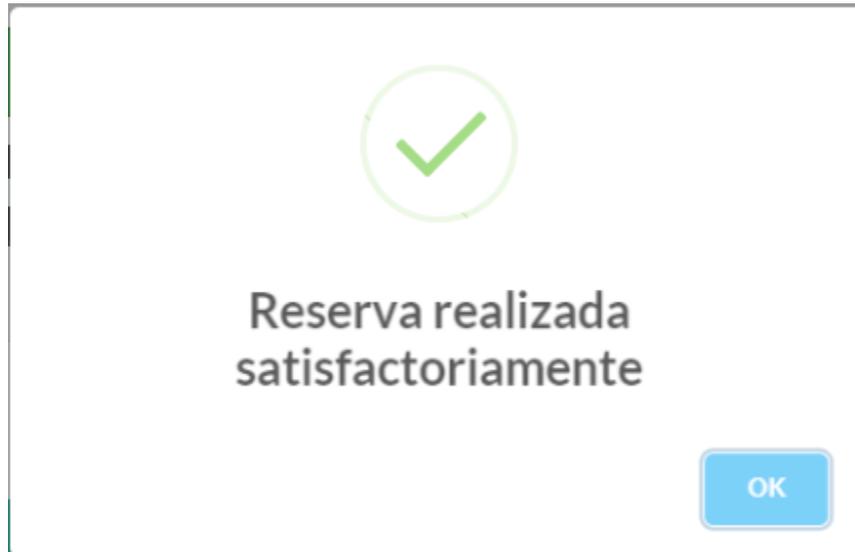
Fecha de final

Motivo o razón

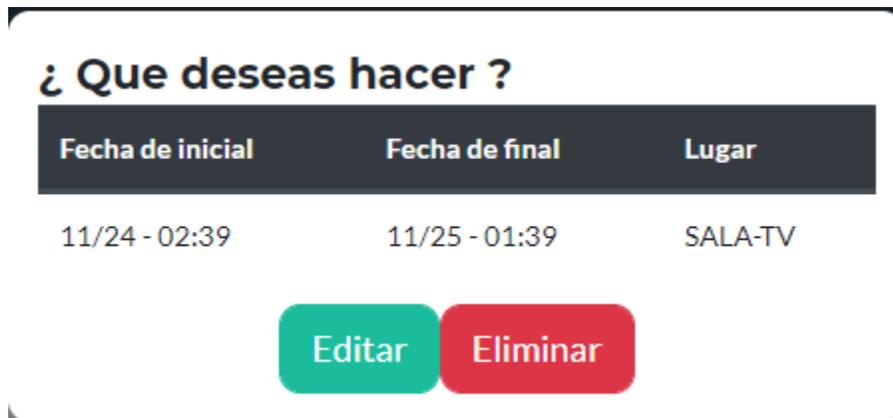
Lugar

Reservar

Al clicar en Reservar se muestra un mensaje de éxito como se muestra a continuación.



Las reservas pueden ser editadas o eliminadas por el estudiante que las creó, siempre y cuando no haya comenzado o haya terminado.



Editar: Solo se permite editar las fechas.

Solo se pueden editar las fechas ×

Fecha de inicial	Fecha de final
11/24 - 02:39	11/25 - 01:39

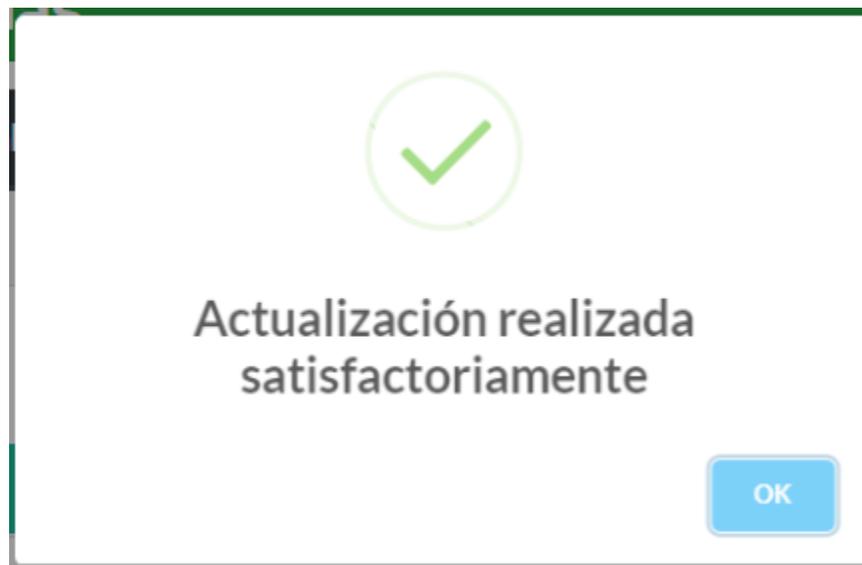
Fecha inicial Actual: 

Fecha Final Actual: 

Motivo o razón

Lugar

Actualización exitosa.



Eliminar: Se visualiza un modal de confirmación para borrar la reserva realizada.

¿ Desea Borrar esta reserva ? Brayan Erazo ×

Fecha de inicial	Fecha de final	Lugar	Motivo
11/24 - 02:39	11/25 - 01:39	SALA-TV	ACADEMICO

Eliminar

Borrado exitoso.



Borrado realizado satisfactoriamente

OK

Labores: El estudiante logueado solo puede visualizar las labores asignadas por un administrador.

Labores asignadas				
Identificacion	Nombre	Espacio	Descripcion	Frecuencia
12345	Camilo Andres Erazo	SALA-TV	Limpiar la sala	2 días por semana

Materias: El estudiante logueado solo puede visualizar las materias (Para el registro de ellas, usar la aplicación móvil).

HJU SANTA MARIA				MENU	PERFIL	LOGOUT
Horarios materias						
Nombre materia	Dia	Hora inicial	Hora Final			
Dinamicos II	LUNES	12	13			
Etica Profesional	JUEVES	18	20			
Etica Profesional	MARTES	18	20			
Moviles	MIERCOLES	18	20			
Moviles	LUNES	18	20			
Operativos	SABADO	08	11			

Inasistencias: El estudiante logueado puede ver la lista de inasistencia y registrar una nueva inasistencia.

Inasistencias realizadas

Nombre	Fecha Inasistencia	Hora Inasistencia	Motivo
Brayan	2020/11/25	DESAYUNO	PERSONAL

Registrar nueva inasistencia

Botón registrar nueva inasistencia: Tener en cuenta que la fecha que se ingrese debe ser válida (Que no haya pasado y que sea una hora antes del horario de alimentación).

Llena todos lo campos para Crear una Inasistencia Camilo Andres

Nota: la fecha debe tener por lo menos una hora de diferecia con respecto a la hora de alimentación

Fecha Inasistencia Motivo o razón

'11/2020 10:09 a. m.  ACADEMICO

Horario

ALMUERZO Crear Inasistencia

Inasistencia realizada satisfactoriamente.



Alerta al cerrar sesión.

