

GRADO EN INGENIERÍA INFORMÁTICA EN GESTIÓN Y SISTEMAS DE LA INFORMACIÓN

TRABAJO FIN DE GRADO

***ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN
SISTEMA DE GESTIÓN DE USUARIOS Y CITAS
PARA EL SERVICIO DE ATENCIÓN DIURNA
IREKIA***

Alumno: Martínez Tadeo, Borja

Directora (1): Blanco Jauregui, Begoña

Director (2): Sainz de Murieta Mangado, Joseba

Curso: 2019 - 2020

Fecha: Bilbao, junio 2020





Resumen: Mediante este Trabajo de Fin de Grado se ha pretendido llevar una mejora del sistema de gestión del Servicio de Atención Diurna IREKIA, perteneciente a la asociación sin animo de lucro Lagun Artean, situada en la calle Juan Urquijo de Bilbao, en el barrio de Deusto. El Servicio de Atención Diurna IREKIA, ofrece distintos servicios: servicio de café-calor, servicio de ducha, servicio de lavandería y clases de castellano.

Actualmente, el sistema no es eficiente y tienen muchos problemas, entre los cuales, esta que hay problemas con las colas que se forman al exterior del local o que no todos los usuarios pueden usar el servicio. Además, los recursos son escasos, ya que solo existen 3 duchas en el local y 18 turnos de ducha disponibles por día. Esto hace que los usuarios tengan que ir a hacer cola con mucho tiempo de antelación y así asegurarse un puesto en la ducha, lo que hace que genere largas colas y un gran impacto visual y acústico. Esto hace que los vecinos que viven al lado se quejen, sobretodo los días de lluvia donde los usuarios se resguardan en un garaje contiguo y debido al eco, el ruido se incrementa.

Por otra parte, la entrada al local se hace lenta ya que los administradores tienen que apuntar en un cuaderno los nombres de todos los usuarios que van a usar el servicio. A través de un sistema de reservas, los usuarios podrían reservar los servicios con anterioridad. De esta forma, los usuarios no tendrían que hacer colas para acceder al local, ya que tienen los servicios reservados.

Para que los usuarios puedan usar el servicio, tendrán que registrarse y así los administradores llevar un control sobre quien usa el servicio. Debido a que los usuarios no conocen o conocen muy poco el castellano, se ha pretendido que sea lo más accesible posible para fomentar el uso de la aplicación.

Palabras clave: sistema de gestión, asociación sin animo de lucro.



Laburpena: Gradu Amaierako Lan honen bidez, Irekia Eguneko Arreta Zerbitzuaren kudeaketa-sistema hobetu nahi izan da. Zerbitzu hori Lagun Artean irabazi-asmorik gabeko elkartearena da, eta Bilboko Juan Urkixo kalean dago, Deustu auzoan. Irekia Eguneko Arreta Zerbitzuak hainbat zerbitzu eskaintzen ditu: kafe-bero zerbitzua, dutxa zerbitzua, garbitegi zerbitzua eta gaztelania eskolak.

Gaur egun, sistema ez da eraginkorra eta arazo asko dituzte, besteak beste, loka-laren kanpoaldean sortzen diren ilarekin arazoak daudela edo erabiltzaile guztiekin ezin dutela zerbitzua erabili. Gainera, baliabide gutxi daude, lokalean 3 dutxa eta eguneko 18 dutxa txanda bakarrik baitaude. Horren ondorioz, erabiltzaileek ilara egin behar izaten dute denbora asko lehenago, dutxan postu bat ziurtatuz, eta ilara luzeak eta inpaktu bisual eta akustiko handia eraginez. Horren ondorioz, ondoan bizi diren bizilagunak kexatu egiten dira, batez ere euri-egunetan, erabiltzaileak ondoko garaje batean babesten baitira, eta oihartzuna dela eta, zarata areagotu egiten da.

Bestalde, lokalerako sarrera motela da, administratzailaik koaderno batean idatzi behar baitituzte zerbitzua erabiliko duten erabiltzaile guztien izenak. Erreserba-sistema baten bidez, erabiltzaileek aldez aurretik erreserbatu ahal izango lituzkete zerbitzuak. Horrela, erabiltzaileek ez lukete ilararik egin beharko lokalera sartzeko, zerbitzuak erreserbatuta baitituzte.

Erabiltzaileek zerbitzua erabili ahal izateko, erregistratu egin beharko dira eta, horrela, administratzailaik zerbitzua erabiltzen duenaren gaineko kontrola era-man beharko dute. Erabiltzaileek gaztelania oso gutxi edo oso gutxi ezagutzen dutenez, aplikazioaren erabilera sustatzeko ahalik eta eskuragarriena izatea lortu nahi izan da.

Gako-hitzak: kudeaketa sistema, irabazi asmorik gabejo elkartea.



Abstract: The aim of this end-of-degree project is to improve the management system of the IREKIA Day Care Service, which belongs to the non-profit association Lagun Artean, located in Juan Urquijo Street in Bilbao, in the Deusto district. The IREKIA Day Care Service offers different services: coffee and heat service, shower service, laundry service and Spanish classes.

Currently, the system is not efficient and they have many problems, among which, there are problems with the queues that are formed outside the premises or that not all users can use the service. In addition, resources are scarce, as there are only 3 showers in the premises and 18 shower shifts available per day. This means that users have to queue up well in advance to secure a place in the shower, which generates long queues and a great visual and acoustic impact. This makes the neighbours living next door complain, especially on rainy days when users take shelter in an adjacent garage and due to the echo, the noise increases.

On the other hand, the entrance to the premises becomes slow because the administrators have to write down in a notebook the names of all the users that are going to use the service. Through a reservation system, users could book the services in advance. In this way, users would not have to queue to access the premises, as they have the services reserved.

In order for users to use the service, they will have to register so that the administrators can keep track of who is using the service. Because users do not know or know very little Spanish, we have tried to make it as accessible as possible to encourage the use of the application.

Keywords: management system, non-profit association.





Índice general

1. Introducción	14
1.1. Definiciones, acrónimos y abreviaturas	15
1.2. Origen del proyecto	15
1.3. Justificación	17
1.4. Planteamiento del problema	17
2. Planteamiento inicial	21
2.1. Objetivos	21
2.2. Arquitectura	22
2.2.1. Cliente - Servidor	22
2.2.2. Modelo - Vista - Controlador	24
2.3. Ciclo de vida	24
2.4. Herramientas	25
2.5. Alcance	30
2.5.1. Estructura de Descomposición del Trabajo	30
2.5.2. Descripción de las tareas	31
2.6. Planificación temporal	37
2.6.1. Diagrama de Gantt	38
2.7. Análisis de alternativas	39
2.7.1. Alojamiento de la aplicación web	39
2.7.2. Bases de Datos	40
2.8. Evaluación económica	42
2.8.1. Hardware	42
2.8.2. Software	42
2.8.3. Mano de obra	43
2.8.4. Otros gastos	43
2.8.5. Gastos totales	43
2.9. Gestión de riesgos	44
2.9.1. Enfermedad	44
2.9.2. Caída de Overleaf	44
2.9.3. Caída de Microsoft Azure	45
2.9.4. Ataques cibernéticos	45
2.9.5. Mal diseño del proyecto	45
2.9.6. Falta de tiempo para realizar pruebas	46
2.9.7. Rotura del equipo de trabajo	46
2.9.8. No cumplir con la LOPD/RGPD	46
2.9.9. El cliente abandone el proyecto	47
2.9.10. Confinamiento por el COVID-19	47
3. Captura de requisitos	48
3.1. Reuniones en Lagun Artean	48
3.2. Casos de Uso	51
3.2.1. Casos de Uso para cualquier usuario	51
3.2.2. Casos de Uso para los usuarios	51
3.2.3. Casos de Uso para los administradores	52
3.2.4. Jerarquía de Actores	52
3.3. Modelo de Dominio	54



4. Análisis y Diseño	57
4.1. Transformación de Modelo de Dominio a Diagrama de Base de Datos	57
4.2. Tablas de la Base de Datos	58
4.3. Diagrama de clases para la aplicación de escritorio	66
4.4. Códigos QR	67
5. Desarrollo	68
5.1. Diseño de la web	68
5.1.1. Back-end	68
5.1.2. Base de datos	68
5.1.3. Mapa	69
5.1.4. Soporte multiidioma	69
5.1.5. Soporte multipantalla	70
5.2. Diseño de la aplicación	71
5.2.1. Maven	71
5.3. API Google Charts	71
5.4. API Google Zxing	72
5.5. API Webcam-Capture	72
5.6. Ley Orgánica de Protección de Datos personales y Garantía de los Derechos Digitales (LOPDGDD)	72
5.7. HTTPS	73
5.8. Lenguajes utilizados	73
5.8.1. Java	74
5.8.2. HTML 5	74
5.8.3. JavaScript	75
5.8.4. PHP	75
5.9. Problemas y soluciones	75
6. Pruebas unitarias	77
6.1. Pruebas relativas respecto al registro	77
6.2. Pruebas relativas respecto al inicio y cierre de sesión	78
6.3. Pruebas relativas respecto a la gestión de las duchas	79
6.4. Pruebas relativas respecto a la gestión de la lavandería	80
6.5. Pruebas relativas respecto a la gestión de las clases	81
7. Conclusiones y trabajo futuro	82
7.1. ¿Se ha cumplido los objetivos establecidos?	82
7.2. El proyecto previsto y el proyecto finalizado	82
7.2.1. Planificación final	83
7.2.2. Evaluación económica final	84
7.3. Trabajo futuro	84
8. Referencias	85
Anexos	87



A. Anexo A: Casos de Uso extendidos	87
A.1. Registro	87
A.2. Inicio de sesión de usuarios	90
A.3. Cambio de contraseña	92
A.4. Comprobar solicitudes de registro de usuarios	94
A.5. Apuntarse a clases de castellano	97
A.6. Ver ayuda	100
A.7. Pedir cita para las duchas	102
A.8. Pedir cita para la lavandería	104
A.9. Ver citas de la ducha	106
A.10. Ver citas de la lavandería	108
A.11. Ver usuarios	110
A.12. Gestionar clases de castellano	112
A.13. Ver ropa	116
A.14. Ver estadísticas sobre el sistema	118
A.15. Apuntarse al café	120
A.16. Mantenimiento duchas	122
A.17. Ver reservas ducha	124
A.18. Ver reservas lavandería	125
A.19. Ver notificaciones	126
B. Anexo B : Diagramas de Secuencia	127
B.1. Diagrama de Secuencia: Registro	127
B.2. Diagrama de Secuencia: Inicio de Sesión	129
B.3. Diagrama de Secuencia: Cambio de Contraseña	131
B.4. Diagrama de Secuencia: Ver ayuda	132
B.5. Diagrama de Secuencia: Comprobar solicitudes de registro de usuarios	133
B.6. Diagrama de Secuencia: Apuntarse a clase de castellano	134
B.7. Diagrama de Secuencia: Pedir cita para las duchas	136
B.8. Diagrama de Secuencia: Pedir cita para la lavandería	138
B.9. Diagrama de Secuencia: Ver citas ducha	139
B.10. Diagrama de Secuencia: Ver citas lavandería	140
B.11. Diagrama de Secuencia: Ver usuarios	141
B.12. Diagrama de Secuencia: Ver usuarios - Editar Usuario	142
B.13. Diagrama de Secuencia: Ver usuarios - Borrar Usuario	144
B.14. Diagrama de Secuencia: Gestionar clases de castellano - Añadir Clase	145
B.15. Diagrama de Secuencia: Gestionar clases de castellano - Eliminar Clase	146
B.16. Diagrama de Secuencia: Gestionar clases de castellano - Matricular Alumno	147
B.17. Diagrama de Secuencia: Gestionar clases de castellano - Desmatricular Alumno	149
B.18. Diagrama de Secuencia: Ver ropa	151
B.19. Diagrama de Secuencia: Ver estadísticas del sistema	152
B.20. Diagrama de Secuencia: Apuntarse al café	154
B.21. Diagrama de Secuencia: Mantenimiento duchas	157
B.22. Diagrama de Secuencia: Ver reservas ducha	158
B.23. Diagrama de Secuencia: Ver reservas lavandería	159



Índice de figuras

1.	Calle donde se encuentra los locales de Lagun Arteen	14
2.	Entrada al centro de atención diurna	18
3.	Garaje contiguo al centro de atención diurna	19
4.	Cliente - Servidor	22
5.	Plano Lagun Arteen	23
6.	Modelo - Vista - Controlador	24
7.	Ciclo de vida en cascada	25
8.	Logo de Google Drive	25
9.	Logo de Overleaf	25
10.	Logo de Gantt Project	26
11.	Logo de Cacoo	26
12.	Logo de IntelliJ IDEA	26
13.	Logo de Microsoft Azure	26
14.	Logo de MySQL	27
15.	Logo de phpMyAdmin	27
16.	Logo de Apache	27
17.	Logo de Visual Paradigm	27
18.	Logo de Diagrams.net	28
19.	Logo de Google Charts	28
20.	Logo de Google Zxing	29
21.	Logo de Maven	29
22.	Estructura de Descomposición del Trabajo	30
23.	Diagrama de Gantt	38
24.	Digital Ocean	39
25.	Node.js + Express	39
26.	MongoDB	40
27.	Firebase	41
28.	Árbol de problemas de las reuniones	48
29.	Árbol de problemas de las reuniones	49
30.	Prototipo Página Web	50
31.	Diagrama de Casos de Uso para cualquier usuario	51
32.	Diagrama de Casos de Uso para los usuarios	51
33.	Diagrama de Casos de Uso para los administradores	52
34.	Jerarquía de Actores	52
35.	Modelo de Dominio	54
36.	Diagrama de Bases de Datos	57
37.	Tabla Usuario	58
38.	Tabla Administrador	59
39.	Tabla Clases	60
40.	Tabla Apuntado	61
41.	Tabla Matricula	61
42.	Tabla Ducha	62
43.	Tabla Duchan	63
44.	Tabla InfoLavanderia	64
45.	Tabla InfoCafe	65
46.	Diagrama de Bases de Datos	66
47.	Ejemplo de código QR	67



48.	Donde estamos	69
49.	Soporte multiidioma	70
50.	Soporte multipantalla	70
51.	Java	74
52.	HTML (HyperText Markup Language)	74
53.	JavaScript	75
54.	PHP	75
55.	Caso de uso extendido: Registro	87
56.	Página principal Web	88
57.	Registro Web	88
58.	Completar registro Web	89
59.	Caso de uso extendido: Inicio de sesión de usuarios	90
60.	Inicio sesión Web	91
61.	Caso de uso extendido: Ver usuarios	92
62.	Cambio Contraseña Web	93
63.	Aviso cambio contraseña Web	93
64.	Caso de uso extendido: Comprobar solicitudes de registro de usuarios	94
65.	Panel administrador Web	95
66.	Registro final Web	95
67.	Editar usuario Web	96
68.	Usuario creado Web	96
69.	Caso de uso extendido: Apuntarse a clases de castellano	97
70.	Panel usuario Web	98
71.	Lista clases Web	98
72.	Aviso clases llenas Web	99
73.	Aviso clases llenas Web	99
74.	Caso de uso extendido: Ver ayuda	100
75.	Ayuda Web	101
76.	Caso de uso extendido: Pedir cita para las duchas	102
77.	Calendario ducha Web	103
78.	Horario ducha Web	103
79.	Caso de uso extendido: Pedir cita para la lavandería	104
80.	Calendario lavandería Web	105
81.	Caso de uso extendido: Ver citas de la ducha	106
82.	Lista ducha Web	107
83.	Caso de uso extendido: Ver citas de la lavandería	108
84.	Lista Lavandería Web	109
85.	Caso de uso extendido: Ver usuarios	110
86.	Lista usuarios Web	111
87.	Caso de uso extendido: Gestionar clases de castellano	112
88.	Menú gestión clases Web	113
89.	Menú matricula Web	113
90.	Menú clases Web	114
91.	Matricular Web	114
92.	Añadir clase Web	114
93.	Desmatricular alumno Web	115
94.	Eliminar clase Web	115
95.	Caso de uso extendido: Ver ropa	116
96.	Lista Ropa Web	117



97.	Caso de uso extendido: Ver estadísticas sobre el sistema	118
98.	Estadísticas	119
99.	Caso de uso extendido: Apuntarse al café	120
100.	Escaneo Código QR	121
101.	Caso de uso extendido: Mantenimiento duchas	122
102.	Mantenimiento Duchas	123
103.	Caso de uso extendido: Mantenimiento duchas	124
104.	Caso de uso extendido: Ver reservas lavandería	125
105.	Caso de uso extendido: Ver notificaciones	126
106.	Diagrama de Secuencia: Registro	127
107.	Diagrama de Secuencia: Inicio Sesión	129
108.	Diagrama de Secuencia: Cambio de Contraseña	131
109.	Diagrama de Secuencia: Ayuda	132
110.	Diagrama de Secuencia: Comprobar solicitudes de registro de usuarios	133
111.	Diagrama de Secuencia: Apuntarse a clase de castellano	134
112.	Diagrama de Secuencia: Pedir cita para las duchas	136
113.	Diagrama de Secuencia: Pedir cita para la lavandería	138
114.	Diagrama de Secuencia: Ver citas ducha	139
115.	Diagrama de Secuencia: Ver citas lavandería	140
116.	Diagrama de Secuencia: Ver usuarios	141
117.	Diagrama de Secuencia: Ver usuarios - Editar Usuario	142
118.	Diagrama de Secuencia: Ver usuarios - Borrar Usuario	144
119.	Diagrama de Secuencia: Gestionar clases de castellano - Añadir Clase	145
120.	Diagrama de Secuencia: Gestionar clases de castellano - Eliminar Clase	146
121.	Diagrama de Secuencia: Gestionar clases de castellano - Matricular Alumno	147
122.	Diagrama de Secuencia: Gestionar clases de castellano - Desmatricular Alumno	149
123.	Diagrama de Secuencia: Ver ropa	151
124.	Diagrama de Secuencia: Ver estadísticas del sistema	152
125.	Diagrama de Secuencia: Apuntarse al café	154
126.	Diagrama de Secuencia: Mantenimiento duchas	157
127.	Diagrama de Secuencia: Ver reservas ducha	158
128.	Diagrama de Secuencia: Ver reservas lavandería	159

Índice de tablas

1.	Paquete de trabajo: Reuniones Lagun Artean	31
2.	Paquete de trabajo: Realizar el DOP	31
3.	Paquete de trabajo: Captura de requisitos	32
4.	Paquete de trabajo: Seleccionar herramientas	32
5.	Paquete de trabajo: Casos de Uso	32
6.	Paquete de trabajo: Modelo de Dominio	33
7.	Paquete de trabajo: Transformación de Modelo de Dominio a BBDD	33
8.	Paquete de trabajo: Diagramas de Secuencia	33



9.	Paquete de trabajo: Instalación de las herramientas	34
10.	Paquete de trabajo: Instalación de la Base de Datos	34
11.	Paquete de trabajo: Implementar funcionalidades comunes	34
12.	Paquete de trabajo: Implementar funcionalidades administrador	35
13.	Paquete de trabajo: Implementar funcionalidades usuario	35
14.	Paquete de trabajo: Realizar plan de pruebas	35
15.	Paquete de trabajo: Verificar pruebas	36
16.	Paquete de trabajo: Realizar memoria del TFG	36
17.	Paquete de trabajo: Realizar los manuales de uso	36
18.	Planificación temporal de las tareas	37
19.	Precio y vida útil del hardware	42
20.	Gastos totales del proyecto	43
21.	Riesgo 1: Enfermedad	44
22.	Riesgo 2: Caída de Overleaf	44
23.	Riesgo 3: Caída de Microsoft Azure	45
24.	Riesgo 4: Ataques cibernéticos	45
25.	Riesgo 5: Mal diseño del proyecto	45
26.	Riesgo 6: Falta de tiempo para realizar pruebas	46
27.	Riesgo 7: Rotura del equipo de trabajo	46
28.	Riesgo 8: No cumplir con la LOPD/RGPD	46
29.	Riesgo 9: El cliente abandone el proyecto	47
30.	Riesgo 10: Cuarentena o confinamiento por el COVID-19	47
31.	Pruebas relativas respecto al registro	77
32.	Pruebas relativas respecto al inicio y cierre de sesión	78
33.	Pruebas relativas respecto a la gestión de las duchas	79
34.	Pruebas relativas respecto a la gestión de la lavandería	80
35.	Pruebas relativas respecto a la gestión de las clases	81
36.	Planificación final de las tareas	83
37.	Gastos totales del proyecto	84



1. Introducción

En la actualidad, los sistemas informáticos de gestión se utilizan en la gran mayoría de organizaciones, desde empresas a organizaciones sin ánimo de lucro. Estos sistemas sirven para realizar una mejor gestión de la organización, desde tener un control de los proveedores hasta guardar información de los clientes de una empresa.

Muchas organizaciones no disponen de este tipo de sistemas de gestión o disponen de un sistema de gestión muy básico. Algunas de las razones por las que no usan este tipo de sistemas suelen ser: porque son caros, no se tiene experiencia en el uso de este tipo de sistemas, o básicamente, las organizaciones han funcionado bien sin este tipo de sistemas. Pero a lo largo del tiempo, se han ido detectando dificultades para gestionar distintos servicios.

El Centro de Acogida de Lagun Artean está situado en la calle Juan Urquijo, 5 en Bilbao, en el barrio de Deusto, es una asociación sin ánimo de lucro vinculada a la Iglesia y a Caritas de Bizkaia que ayuda a la integración social de las personas sin hogar y personas en zona de exclusión social. El Centro de Acogida de Lagun Artean tiene diferentes servicios:

- Centro de Noche/Albergue
- Centro de Día Kide
- Viviendas tuteladas
- Servicio de Atención Diurna IREKIA



Figura 1: Calle donde se encuentran los locales de Lagun Artean

El último de estos servicios, el Servicio de Atención Diurna IREKIA, presenta dificultades para gestionar el acceso al servicio que oferta. Dentro del proceso interno de innovación surgió la idea de la creación de una aplicación informática



que pueda ser utilizada en el móvil y ordenador para gestionar las prestaciones del servicio.

1.1. Definiciones, acrónimos y abreviaturas

- **ISF-MGI** : Ingeniería Sin Fronteras País Vasco - Euskal Herriko Mugarik Gabeko Ingeniaritza.
- **ONGD**: Organización No Gubernamental para el Desarrollo.
- **TFG**: Trabajo de Fin de Grado.
- **TFM**: Trabajo de Fin de Master.
- **PFC**: Proyecto Fin de Carrera.
- **CRUE**: Conferencia de Rectores de Universidades Españolas.
- **IDE**: Se trata de un entorno de desarrollo integrado , en inglés, *Integrated Development Environment*.
- **SGBD**: Sistema de Gestión de Bases de Datos.
- **HTML**: HyperText Markup Language.
- **CSS**: Se le denomina hoja de estilos en cascada, en inglés, *Cascading Style Sheets*.
- **EDT**: Estructura de Descomposición del Trabajo.
- **QR**: Es un código de respuesta rápida, en inglés *Quick Response code*.
- **Back-end**: Es la parte del desarrollo que se corresponde con la parte lógica de cualquier aplicación, esta normalmente esta alojada en la parte del servidor. El back-end no dispone de parte gráfica por lo que el usuario no lo puede ver.
- **Front-end**: Es la parte gráfica de cualquier aplicación. Se encarga de interactuar con el usuario para después el back-end hacer toda la manipulación.

1.2. Origen del proyecto

Este proyecto nace a partir de la necesidad del Centro de Acogida Lagun Artean de mejorar su Servicio de Atención Diurna IREKIA. El Centro de Acogida Lagun Artean de pone en contacto con ISF-MGI para estudiar una posible solución de mejora del servicio.

ISF-MGI es una ONGD nacida en el año 1994 formada por alumnos y profesores de diferentes disciplinas que desarrolla actividad en la universidad en el marco de la Educación para la Transformación Social y la Cooperación Internacional. ISF-MGI lleva ya veinticinco años tratando de establecer puentes entre la tecnología y las necesidades de personas y colectivos con el Desarrollo Humano.



Desde el curso 2006/2007 lleva en marcha el Programa de Proyectos de Fin de Carrera en Cooperación al Desarrollo, la cual es una de las formas de establecer puentes entre la ingeniería, la universidad y necesidades sociales. A través de este programa, ISF-MGI ha coordinado multitud de TFG, TFM y PFC.

Tras conversaciones entre el Centro de Acogida Lagun Artean e ISF-MGI, se traslada la idea del proyecto a la profesora Begoña Blanco, directora del mismo y tras la buena acogida por su parte, se traslada la idea al alumnado de cuarto curso del Grado de Ingeniería Informática en Gestión y Sistemas de la Información.

Se plantea la idea de poder desarrollar un proyecto para ayudar en la creación de un sistema informático para gestionar los distintos servicios que oferta el Servicio de Atención Diurna IREKIA. Hasta entonces, el servicio carecía de sistemas informáticos para registrar los usuarios que utilizaban el servicio. Solo usan como sistema informático la base de datos del ayuntamiento para las personas que hacen noche en el centro de acogida, para las demás funciones no existe sistema informático.

El Servicio de Atención Diurna IREKIA se define como un servicio de baja intensidad de apoyo destinado a 20 personas en exclusión social. Al tratarse de un servicio de baja exigencia y de uso temporal, solo funciona de lunes a viernes de 9 a 16 horas. Este servicio facilita un espacio para poder resguardarse durante el día y cubre las necesidades básicas como son: desayuno y café-calor, higiene personal, lavandería, consigna y ropa de urgencia.

Además, a las personas usuarias del servicio, se les facilita apoyo profesional en el centro para: itinerarios de inserción social, información y orientación sobre sus derechos, realización de gestiones administrativas, diferentes actividades formativas básicas como clases de castellano y habilidades sociales.

A la organización le ha llevado a la creación de un nuevo sistema de gestión atajar una serie de problemas que se iban acumulando:

- Evitar colas que se generan en la entrada del local
- Reducir el impacto visual y acústico que se genera
- Disminuir el malestar vecinal derivado de las colas que se generan
- Las personas que realizan cursos de formación no pueden llegar a la hora de apertura
- Carencia de recursos humanos en el servicio

También, las mejoras que se esperan tras la creación de la aplicación informática y así gestionar de una manera más eficiente las prestaciones del Servicio de Atención Diurna IREKIA:

- Mejorar y dignificar la accesibilidad al servicio
- Garantizar el acceso a todas las personas sin hogar
- Reducir el malestar de la comunidad



1.3. Justificación

En este apartado hablaremos que tanto este como otros proyectos del mismo tipo se justifican también al entenderlos como una parte fundamental de las funciones de la universidad.

La Ley Orgánica 4/2007, que viene a modificar la Ley 6/2001, de Universidades, en su artículo 92, de la cooperación internacional y la solidaridad establece que:

"Las universidades fomentarán la participación de los miembros de la comunidad universitaria en actividades y proyectos de cooperación internacional y solidaridad. Asimismo, propiciarán la realización de actividades e iniciativas que contribuyan al impulso de la cultura de la paz, el desarrollo sostenible y el respeto al medio ambiente, como elementos esenciales para el progreso solidario."

Este proyecto viene a conectar el trabajo social con la ingeniería informática, dos disciplinas muy distantes, pero que uniéndolas pueden llegar a mejorar un poco más la sociedad. Hasta ahora, no ha habido proyectos reales de estas características. Esta es una gran oportunidad para que alumnos de TFG de próximos años puedan elegir un proyecto de estas características y mostrar sus conocimientos fuera de las aulas.

Como el servicio que oferta Lagun Arteen no tenía hasta ahora un sistema de estas características, se requerirá la máxima participación del cliente y de los usuarios para así hacer el sistema lo más sencillo y eficaz posible.

Todo esto supone un reto muy enriquecedor y me ha animado a elegirlo como Trabajo de Fin de Grado.

1.4. Planteamiento del problema

Este proyecto tenía como objetivo mejorar el servicio que se ofrece en el Servicio de Atención Diurna IREKIA. Partíamos de que en el servicio no se utilizaba ningún sistema informático para controlar el flujo de usuarios que usaban el servicio cada día.

Hasta ahora el servicio no disponía de ningún sistema para controlar el flujo de personas, los usuarios accedían al local y los administradores les apuntaban en un cuaderno su nombre y apellido. Para verificar su edad ya que no ofrecen el servicio a menores de edad, les pedían una identificación ya sea una documentación oficial como un pasaporte o bien una tarjeta que pusiese su nombre, apellidos y edad, ya que muchos de ellos no disponen de documentación.

Cuando los administradores apuntaban los nombres y apellidos en el cuaderno, se les proporcionaba un número que servía para mantener un orden para tomar el café. En el caso de que los usuarios quisieran usar el servicio de ducha, al lado del nombre se les escribía una 'D', así hasta marcar las 18 duchas disponibles cada día. Como en cada servicio de 14:00 a 16:00 solo se podían ofertar 18 turnos de ducha, debido a que solo existen 3 duchas, había gente que se quedaban sin poder utilizar el servicio y solo les quedaba la opción de consumir un



café o poder usar el servicio de lavandería, ya que para estos servicios, no hay un numero máximo de usuarios que puedan usar el servicio.

Para el servicio de lavandería, acudían directamente al puesto de lavandería para pedir cita para traer la ropa al día siguiente, el día la siguiente la entregan y dentro de dos días la recogen del servicio de lavandería. Además de estos servicios, se oferta uno de clases de castellano que funciona a las mañana, pero que no tiene problemas de gestión debido a que el numero de usuarios no es tan alto.

El sistema que tienen implantado presenta diferentes problemas, que con el sistema informático que se planteara a continuación pasaran a eliminarse. Los problemas existentes son:

- Evitar colas: Actualmente se generan muchas colas a la entrada del local (Figura 2) para poder usar los distintos servicios que se ofrecen. Estas colas se forman sobretodo por las personas que desean usar el servicio de ducha. De esta forma, se aseguran tener un turno para la ducha.



Figura 2: Entrada al centro de atención diurna

- Numeración para el café: Los administradores tiene que estar imprimiendo constantemente nuevos numero para el café, ya que muchos usuarios utilizan la picaresca para guardarse los mejores números para días en los que llegan mas tarde al local.
- Reducir impacto visual y acústico: Viene generado por las colas, cuando hay mayor impacto acústico son los los días de lluvia donde los usuarios se tienen que resguardar en el garaje contiguo al local, donde debido al eco el ruido es mayor.



Figura 3: Garaje contiguo al centro de atención diurna

Además, como se puede apreciar en las imágenes (Figura 2) y (Figura 3) no hay un lugar apropiado para resguardarse en los días de lluvia y así reducir el impacto acústico y visual.

- Personas que trabajan o estudian puedan usar el servicio: El servicio lo utilizan multitud de personas, gente que trabaja, gente que esta estudiando etc. Por lo que muchos de ellos no pueden ir todos los días a usar el servicio.
- Carencia de recursos humanos: En el local de Lagun Artean al disponer de pocos recursos humanos, no pueden atender personalmente muchas de las peticiones de los usuarios.

El nuevo sistema informático que se plantea viene a resolver los problemas que tienen en el servicio. El sistema que se plantea viene dividido en dos aplicaciones. Una de ellas, una aplicación web, que permita a los usuarios poder registrarse para que así después puedan pedir cita para la lavandería, ducha y clases de castellano. Además, los administradores a través de esta misma aplicación podrán llevar una gestión de los usuarios que usan el servicio y ver que



usuarios usan el servicio de ducha, lavandería y las clases de castellano. Como funcionalidad adicional y así liberar trabajo a los administradores, podrán recoger diferentes estadísticas del sistema.

La otra aplicación se trata de una aplicación de escritorio, que estaría alojada en un ordenador en la entrada del local. Esta estaría conectada a una webcam, para que los usuarios puedan identificarse mediante un código QR. Cuando los usuarios se identifican, se les imprimirá un ticket que les indicaría el numero del turno del café, y si ese día tienen concertada una cita para la ducha o la lavandería.

Con el nuevo sistema, los diferentes servicios funcionarían de la siguiente forma:

- En el caso de reserva *online* del servicio de ducha, los usuarios acudirían a la hora que tienen asignado el turno de ducha y solo tendrían que dar el nombre para el turno del café. De esta forma, se eliminaría gran parte de la cola que se genera en la parte exterior del establecimiento y personas que realizan cursos de formación tendrían la opción de poder llegar mas tarde y poder hacer uso del servicio.
- Para el servicio del café, los usuarios accederían al establecimiento y con un ordenador y una webcam que habría en la entrada, un usuario acercaría su código QR que le identifica como un usuario del sistema. Tras acercarlo y que un programa informático le identificase, le imprimiría un ticket que fuese valido para el café para ese día en concreto.
- Con el sistema de gestión informatizado, en el servicio de lavandería, los usuarios se ahorrarían un día para pedir cita, ya que la piden *online* y de esta forma solo tienen que volver a acudir cuando les comuniquen que a ropa esta lista.
- En el caso de las clases de castellano, los usuarios se apuntarían a una clase, pero tendrían que ir al local a hacer una pequeña prueba de nivel para que les matriculasen en el nivel correcto.



2. Planteamiento inicial

En este apartado se explicará el planteamiento inicial del proyecto: objetivos, alcance, planificación etc. Además, se expondrán las aplicaciones y herramientas utilizadas en el desarrollo y se realizará una comparación con herramientas similares que se podrían haber utilizado. También, se analizará los riesgos que pueden surgir durante la realización del proyecto y una estimación del coste económico.

2.1. Objetivos

El objetivo general del proyecto, es entregar el sistema de gestión de usuarios al servicio de atención diurna IREKIA. Se entregará con los requisitos establecidos y para ello se han establecido unos puntos guía a cumplir. Además del objetivo general, se han distinguido diferentes objetivos principales en el proyecto.

- **Reducción de las colas:** Con la nueva aplicación web, se le da la oportunidad a los usuarios de reservar un turno para la ducha, lo que significa, que si se llenan todos los huecos destinados para la ducha ese día, habrá 18 personas menos en la cola. Además, como todas las personas que acudan al local tendrán un café, reducirá considerablemente la cola.
- **Reducir el impacto visual y acústico:** Este objetivo viene a ser una continuación de la reducción de las colas. Ahora en los días de lluvia, en el garaje contiguo al local, por lo menos tendría que haber menos usuarios ya que estos estarían usando la propia aplicación para poder reservar el servicio de ducha.
- **Incrementar el numero de usuarios que pueden usar el sistema:** Con el sistema anterior, las personas que trabajan o estudian tenían muy difícil usar el sobretodo el servicio de ducha ya que tenían que acudir con anterioridad a hacer cola. Con la implementación del nuevo servicio, los usuarios que antes no podían hacer cola, podrán reservar una cita para la ducha con anterioridad para así disfrutar del servicio a la hora que les venga bien.
- **Reducir a dos días el servicio de lavandería:** Con la nueva aplicación web, se consigue que un usuario solo tenga que acudir dos días al local de Lagun Artean, el día que acude a entregar la ropa y el día que acude a recogerla. En comparación con el sistema actual, los usuarios tienen que ir un primer día pedir turno, un segundo día a entregarla y dos días después de entregar la ropa, el usuario acude a recogerla.
- **Satisfacer a los miembros de Lagun Artean con el proyecto creado:** Debido a que en Lagun Artean disponen de pocos recursos humanos para administrar el servicio y con los anteriores objetivos cumplidos, los administradores tendrán menos trabajo para así poder ayudar más a los usuarios del servicio. Por ejemplo, ahora no tendrá que haber un administrador en la puerta controlando el acceso, ya que disponen de una aplicación que hace ese trabajo. Si hemos cumplido con los objetivos anteriores habremos cumplido el presente objetivo.



2.2. Arquitectura

En este apartado se hablará sobre la arquitectura que se utilizara en la aplicación web. Este apartado es fundamental para escribir como funcionará la lógica de negocio.

2.2.1. Cliente - Servidor

El tipo de arquitectura en la que nos basamos es en la arquitectura Cliente - Servidor (Figura 4). Esta arquitectura nos permite modelar el software de tal manera que las tareas se reparten entre el servidor y el cliente.

Gracias a esto, un usuario puede acceder a sus datos independientemente de que dispositivo se conecte. Puede conectarse desde un Smartphone, Tablet u ordenador y tener siempre los datos a su alcance.

Los dispositivos del cliente, se conectan mediante el protocolo HTTP al servidor para poder enviar y recibir datos.

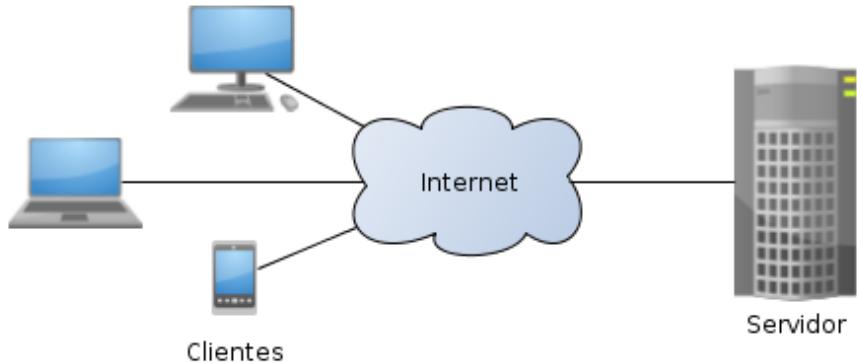


Figura 4: Cliente - Servidor

Aun así, el proyecto es mucho más completo que una arquitectura cliente - servidor. Podemos observar en el siguiente plano del Lagun Artean (Figura 5) el cual se compone de las oficinas y del centro de día, los cuales están separados en la calle.

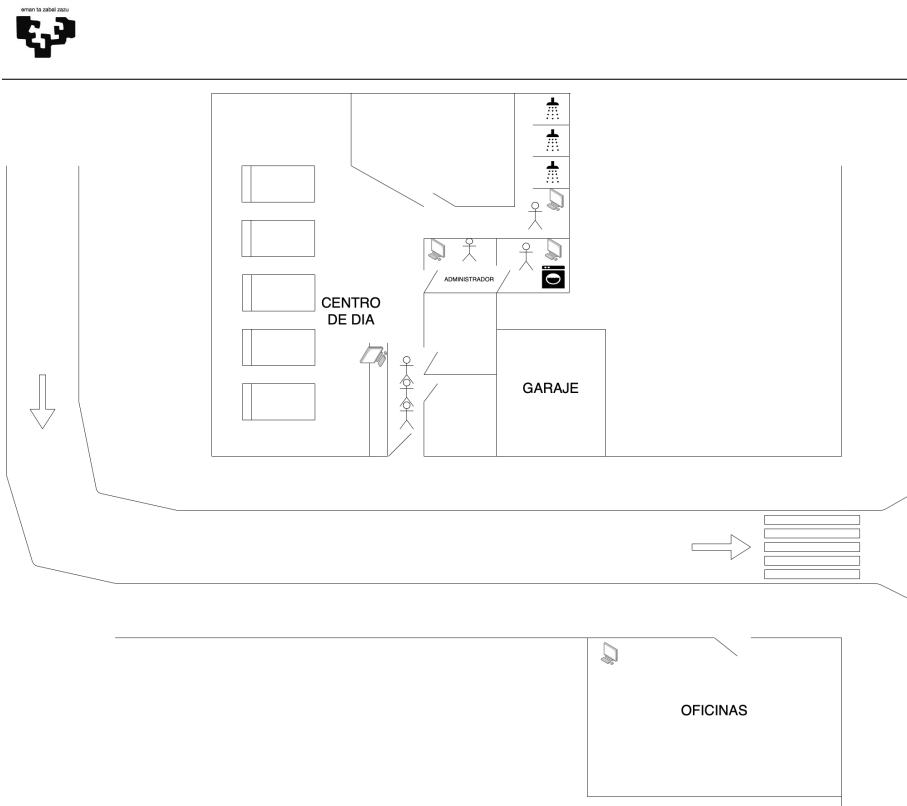


Figura 5: Plano Lagun Artean

El funcionamiento en cada uno de los puestos es el siguiente:

- **Oficinas:** En las oficinas se encuentran los responsables de Lagun Artean, son las personas que se encargan de toda la gestión de la administración. Desde este lugar, podrán ver los usuarios que utilizan los distintos servicios del centro de atención diurna.
- **Centro de día:** Es el lugar donde se ofertan los distintos servicios del centro de atención diurna. Dentro de él, habrá distintos terminales instalados para la gestión de los servicios ofrecidos:
 - **Cola de entrada:** En la cola de entrada se situara un terminal con una cámara web instalada y a su lado una impresora de etiquetas. Cuando entre un usuario, este escaneara en la cámara web su código QR y se le imprimirá un ticket con un turno para el café, un recordatorio de su cita para la ducha o lavandería para ese día.
 - **Administrador:** El administrador es la persona que gestiona el centro de atención diurna, aquí podrá ver y gestionar al igual que en las oficinas los usuarios que usan el servicio.
 - **Duchas:** En la zona de duchas, existirá un terminal donde habrá un voluntario del servicio controlando las personas que tienen reserva para la ducha para cada día.
 - **Lavandería:** Al igual que en la zona de las duchas, habrá un voluntario del servicio que vaya comprobando las citas de la lavandería de cada día y marcando la ropa que ya está lista para recoger.



2.2.2. Modelo - Vista - Controlador

Para el caso de la aplicación para poder detectar el código QR de los usuarios del sistema, usaremos la arquitectura Modelo - Vista - Controlador (Figura 6). Se procederá a utilizar esta arquitectura ya que será una aplicación distinta a la aplicación web. Pero debido a que comparten la misma base de datos, el controlador se conectara al mismo servidor de base de datos que la aplicación web.

Esta arquitectura nos permite separar los datos y la lógica de negocio de la interfaz y el módulo encargado de gestionar los eventos y las comunicaciones. De esta manera, el proyecto está abierto a posibles cambios de plataforma, interfaz, ... sin tener que cambiar el código fuente completamente.

Además, esta arquitectura nos facilita la reutilización de código y por supuesto, la programación estructurada y clara para que lo pueda entender cualquier persona. Esta arquitectura, como bien dice su nombre, se basa en tres pilares:

- **El modelo:** la representación de la información.
- **El controlador:** se encarga de conectar estos dos últimos pilares y gestionar los eventos que sucedan en la interfaz.
- **La vista:** la interfaz de la aplicación.

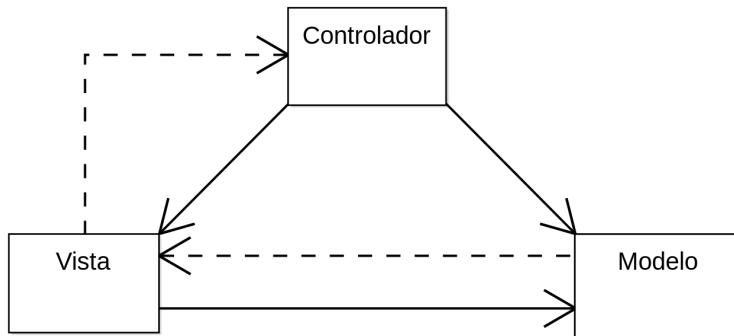


Figura 6: Modelo - Vista - Controlador

2.3. Ciclo de vida

En el desarrollo de software se elige un ciclo de vida desde el momento en que se inicia hasta la finalización. El ciclo de vida tiene unas fases similares en la mayoría de desarrollos.

El ciclo de vida del proyecto, será en cascada (Figura: 7). Se hará uso de este tipo de ciclo de vida, debido a que presenta una flexibilidad y gestión sencilla con las etapas correctamente diferenciadas.



Se comenzará por una captura de los requisitos del proyecto, desde la cual se realizará el posterior análisis. Una vez realizado el análisis con su correspondiente modelo de dominio, se pasará a diseñar e implementar la aplicación. En la fase final, se diseñarán y ejecutarán las pruebas para comprobar el correcto funcionamiento y, si todo es correcto, se presentará al cliente.

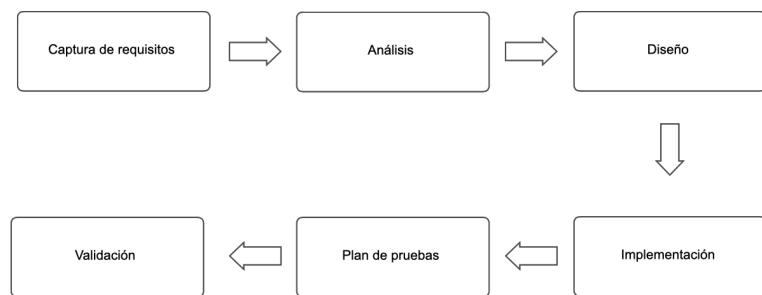


Figura 7: Ciclo de vida en cascada

2.4. Herramientas



Figura 8: Logo de Google Drive

- **Google Drive:** La primera herramienta a la que damos uso es Google Drive. Creada por Google en 2012, nos permite compartir toda la documentación entre los participantes y el cliente, así como editarlos de manera conjunta y guardarlos en la nube.



Figura 9: Logo de Overleaf

- **Overleaf:** Es un editor de LaTeX online, este editor será utilizado para redactar la documentación.



Figura 10: Logo de Gantt Project

- **GanttProject:** GanttProject es un programa de código abierto, con el que podremos administrar y controlar las tareas relacionadas con el proyecto. Nos permite crear jerarquías entre los miembros del grupo, definir límites de tiempo para realizar las tareas o quien se encargará de realizarlas, entre otras muchas.



Figura 11: Logo de Cacoo

- **Cacoo:** Software para la realización de distintos diagramas, entre ellos el diagrama de Estructura de Descomposición del Trabajo.



Figura 12: Logo de IntelliJ IDEA

- **IntelliJ Idea:** Creado por JetBrains, es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos, este IDE se utilizará para desarrollar la correspondiente aplicación web.



Figura 13: Logo de Microsoft Azure



-
- **Microsoft Azure:** Es un servicio en la nube ofrecido por Microsoft. Este servicio se utilizara para alojar la futura aplicación web.



Figura 14: Logo de MySQL

- **MySQL:** Es un sistema de gestión de bases de datos relacional. MySQL será utilizado para la creación de la base de datos. Es gratuito y es fácilmente escalable.



Figura 15: Logo de phpMyAdmin

- **phpMyAdmin:** Es una herramienta que viene en el instalador de XAMPP que sirve para ver de una manera mas visual los cambios realizados en el sistema de gestión de bases de datos de MySQL.



Figura 16: Logo de Apache

- **Apache:** Es un servidor web HTTP para alojar servicios web. Debido a que la web contendrá paginas PHP, este es el servidor ideal para alojar la pagina web.



Figura 17: Logo de Visual Paradigm



-
- **Visual Paradigm:** Es un programa de ordenador que sirve para diseñar diagramas UML. Con él se realizará la Captura de Requisitos y el Análisis y Diseño, creando los Diagramas de Casos de Uso, Modelo de Dominio, Diagramas de Clases y Diagramas de Secuencia.



Figura 18: Logo de Diagrams.net

- **Diagrams.net:** Diagrams.net es una herramienta muy útil a la hora de hacer esquemas, diagramas, planos etc. Se ha utilizado para la realización de esquemas y planos para explicar la arquitectura y ciclo de vida del proyecto.



Figura 19: Logo de Google Charts

- **API Google Charts:** La API de Google Charts es una API para crear gráficos de Google. La API tiene una gran variedad de gráficos a elegir por eso ha sido utilizada para mostrar diversos gráficos a los administradores de Lagun Artean.



Figura 20: Logo de Google Zxing

- **API Google Zxing:** El API Google ZXing, es un proyecto de código abierto de Google, para la lectura de códigos multiformato implementada en Google. Esta librería será utilizada para la lectura de los códigos QR de los usuarios que accedan al servicio de café.
- **API Webcam-Capture:** Este API hace uso del API Google ZXing para capturar las imágenes a partir de cualquier webcam. Esta API integra los drivers necesarios para poder obtener la imagen de una webcam independientemente del ordenador y del sistema operativo que tenga



Figura 21: Logo de Maven

- **Maven:** Maven es una herramienta utilizada para la gestión y construcción de proyectos Java.



2.5. Alcance

En este apartado, se van a explicar las diferentes fases del proyecto. Primero se realizará un diagrama EDT (Estructura de Descomposición del Trabajo), posteriormente se describirán las tareas nombradas en el EDT (descripción, tareas precedentes y salidas) y la duración de cada una de ellas.

2.5.1. Estructura de Descomposición del Trabajo

Respecto a las tareas, se ha desarrollado una Estructura de Descomposición del Trabajo (Figura 22), en la cual se organizan las tareas en distintos campos.

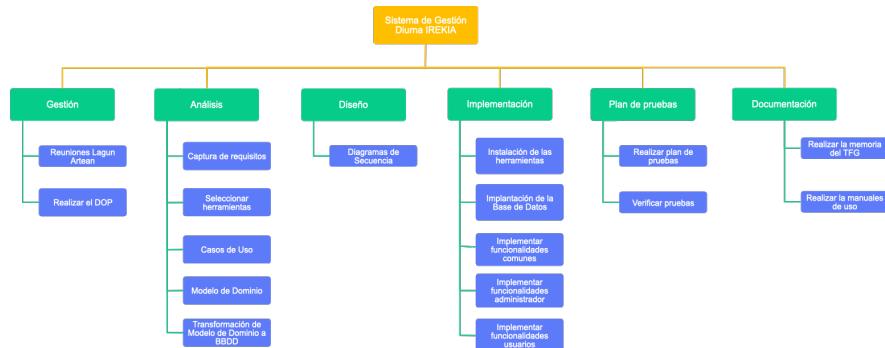


Figura 22: Estructura de Descomposición del Trabajo

- **Gestión:** Consistirá en aclarar todas las dudas posibles acerca del proyecto, dejando claros todos los apartados clave con el cliente para su rea-lización. También se realizará el Documento de Objetivos del Proyecto (DOP).
- **Análisis:** Se elegirán todas las herramientas que usaremos a lo largo del proyecto, la realización de la captura de requisitos, Casos de Uso y Modelo de Dominio para después transformarlo en un diagrama de Base de Datos.
- **Diseño:** En el apartado de diseño, se realizaran los distintos diagramas de secuencia para cada funcionalidad de la aplicación.
- **Implementación:** Se comenzara instalando las herramientas necesarias para su implementación, así como, la implantación de la base de datos y la implementación de las distintas funcionalidades.
- **Plan de pruebas:** Se diseñaran y se comprobarán las pruebas para el correcto funcionamiento de la aplicación.
- **Documentación:** Realizar la documentación necesaria para la memoria del TFG y manuales de uso tanto para la parte de administración como de usuarios.



2.5.2. Descripción de las tareas

1. Gestión

1.1. Paquete de trabajo: Reuniones Lagun Artean

Duración	20 horas
Descripción	Reuniones con los responsables de Lagun Artean. Sirven para establecer las funcionalidades de la aplicación
Tareas precedentes	Ninguna.
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	Ninguna.

Tabla 1: Paquete de trabajo: Reuniones Lagun Artean

1.2. Paquete de trabajo: Realizar el DOP

Duración	24 horas.
Descripción	Realización del DOP. Documento en el que se encuentran recogidas las intenciones y objetivos del proyecto.
Tareas precedentes	Ninguna
Herramientas necesarias	Un ordenador y Overleaf
Entradas	Ninguna
Salidas	El Documento de Objetivos del Proyecto (DOP).

Tabla 2: Paquete de trabajo: Realizar el DOP



2. Análisis

2.1. Paquete de trabajo: Captura de requisitos

Duración	30 horas.
Descripción	Describir los requisitos y funcionalidades, para definir posteriormente los casos de uso.
Tareas precedentes	Ninguna
Herramientas necesarias	Un ordenador y Overleaf
Entradas	Ninguna.
Salidas	Documento de requisitos.

Tabla 3: Paquete de trabajo: Captura de requisitos

2.2. Paquete de trabajo: Seleccionar herramientas

Duración	20 horas.
Descripción	Se han definido las herramientas que se van a utilizar durante la elaboración del proyecto.
Tareas precedentes	Ninguna
Herramientas necesarias	Un ordenador.
Entradas	Ninguna
Salidas	Informe de las distintas herramientas existentes

Tabla 4: Paquete de trabajo: Seleccionar herramientas

2.3. Paquete de trabajo: Casos de Uso

Duración	35 horas
Descripción	En este paquete de trabajo se realizara el diagrama de Casos de Uso y de Casos de Uso extendidos
Tareas precedentes	Captura de requisitos
Herramientas necesarias	Un ordenador, draw.io, Google Drive y Overleaf.
Entradas	Documento de requisitos.
Salidas	Casos de Uso

Tabla 5: Paquete de trabajo: Casos de Uso



2.4. Paquete de trabajo: Modelo de Dominio

Duración	20 horas
Descripción	Elaboración del Modelo de Dominio, describe las entidades, atributos así como las relaciones que hay entre entidades
Tareas precedentes	Casos de Uso.
Herramientas necesarias	Un ordenador, draw.io, Google Drive y Overleaf
Entradas	Ninguna
Salidas	Modelo de Dominio

Tabla 6: Paquete de trabajo: Modelo de Dominio

2.5. Paquete de trabajo: Transformación de Modelo de Dominio a BBDD

Duración	10 horas
Descripción	Elaboración del diagrama de BBDD a partir del Modelo de Dominio
Tareas precedentes	Modelo de Dominio
Herramientas necesarias	Un ordenador, draw.io, Google Drive y Overleaf
Entradas	Modelo de Dominio
Salidas	Diagrama de BBDD

Tabla 7: Paquete de trabajo: Transformación de Modelo de Dominio a BBDD

3. Diseño

3.1. Paquete de trabajo: Diagramas de Secuencia

Duración	25 horas
Descripción	Realización de los Diagrama de Secuencia correspondientes a cada Casos de Uso.
Tareas precedentes	Transformación de Modelo de Dominio a BBDD y Casos de Uso
Herramientas necesarias	Un ordenador, draw.io, Google Drive y Overleaf
Entradas	Casos de Uso
Salidas	Diagramas de Secuencia

Tabla 8: Paquete de trabajo: Diagramas de Secuencia



4. Implementación

4.1. Paquete de trabajo: Instalación de las herramientas

Duración	15 horas
Descripción	Instalar las diferentes herramientas para el desarrollo que se van a utilizar en el proyecto
Tareas precedentes	Ninguna
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	Ninguna

Tabla 9: Paquete de trabajo: Instalación de las herramientas

4.2. Paquete de trabajo: Instalación de la Base de Datos

Duración	15 horas
Descripción	Instalación de la Base de Datos que se va a utilizar durante el proyecto, así como, la creación de tablas y relaciones
Tareas precedentes	Instalación de las herramientas
Herramientas necesarias	Un ordenador
Entradas	Transformación del Modelo de Dominio a BBDD
Salidas	Ninguna

Tabla 10: Paquete de trabajo: Instalación de la Base de Datos

4.3. Paquete de trabajo: Implementar funcionalidades comunes

Duración	50 horas
Descripción	Implementación de las funcionalidades comunes de la aplicación.
Tareas precedentes	Instalación de la Base de Datos e Instalación de las herramientas
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	Ninguna

Tabla 11: Paquete de trabajo: Implementar funcionalidades comunes



4.4. Paquete de trabajo: Implementar funcionalidades administrador

Duración	50 horas
Descripción	Implementación de las funcionalidades que va a realizar el administrador.
Tareas precedentes	Instalación de la Base de Datos e Instalación de las herramientas
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	Ninguna

Tabla 12: Paquete de trabajo: Implementar funcionalidades administrador

4.5. Paquete de trabajo: Implementar funcionalidades usuario

Duración	50 horas
Descripción	Implementación de las funcionalidades que van a realizar los usuarios.
Tareas precedentes	Instalación de la Base de Datos e Instalación de las herramientas
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	La aplicación final

Tabla 13: Paquete de trabajo: Implementar funcionalidades usuario

5. Plan de pruebas

5.1. Paquete de trabajo: Realizar plan de pruebas

Duración	30 horas
Descripción	El plan de pruebas para comprobar el funcionamiento de la aplicación
Tareas precedentes	Ninguna
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	Ninguna

Tabla 14: Paquete de trabajo: Realizar plan de pruebas



5.2. Paquete de trabajo: Verificar pruebas

Duración	30 horas
Descripción	Verificar las pruebas que se han establecido en el plan de pruebas
Tareas precedentes	Realizar plan de pruebas
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	La aplicación final

Tabla 15: Paquete de trabajo: Verificar pruebas

6. Documentación

6.1. Paquete de trabajo: Realizar memoria del TFG

Duración	115 horas
Descripción	En este paquete de trabajo se realizara la memoria del Trabajo de Fin de Grado
Tareas precedentes	Ninguna
Herramientas necesarias	Un ordenador y Overleaf
Entradas	Ninguna
Salidas	Memoria del TFG

Tabla 16: Paquete de trabajo: Realizar memoria del TFG

6.2. Paquete de trabajo: Realizar los manuales de uso

Duración	5 horas
Descripción	En este paquete de trabajo, se realizaran los manuales de usuario para el cliente de la aplicación
Tareas precedentes	Verificar pruebas
Herramientas necesarias	Un ordenador
Entradas	Ninguna
Salidas	Manuales de uso de la aplicación

Tabla 17: Paquete de trabajo: Realizar los manuales de uso



2.6. Planificación temporal

Tras decidir y explicar las tareas que se van a realizar durante el proyecto, antes de organizar los tiempos, indicamos los tiempos en la siguiente tabla para cada tarea.

Tarea	Estimación
1. Gestión	44 horas
1.1. Reuniones Lagun Artean	20 horas
1.2. Realizar el DOP	24 horas
2. Análisis	115 horas
2.1. Captura de requisitos	30 horas
2.2. Seleccionar herramientas	20 horas
2.3. Casos de Uso	35 horas
2.4. Modelo de Dominio	10 horas
2.5. Transformación del Modelo a BBDD	20 horas
3. Diseño	25 horas
3.1. Diagramas de Secuencia	25 horas
4. Implementación	180 horas
4.1. Instalación de las herramientas	15 horas
4.2. Instalación de la Base de Datos	15 horas
4.3. Implementar funcionalidades comunes	50 horas
4.4. Implementar funcionalidades administrador	50 horas
4.5. Implementar funcionalidades usuario	50 horas
5. Plan de pruebas	60 horas
5.1. Realizar plan de pruebas	30 horas
5.2. Verificar pruebas	30 horas
6. Documentación	120 horas
6.1. Realizar memoria del TFG	115 horas
6.2. Realizar los manuales de uso	5 horas
Horas totales	544 horas

Tabla 18: Planificación temporal de las tareas

2.6.1. Diagrama de Gantt

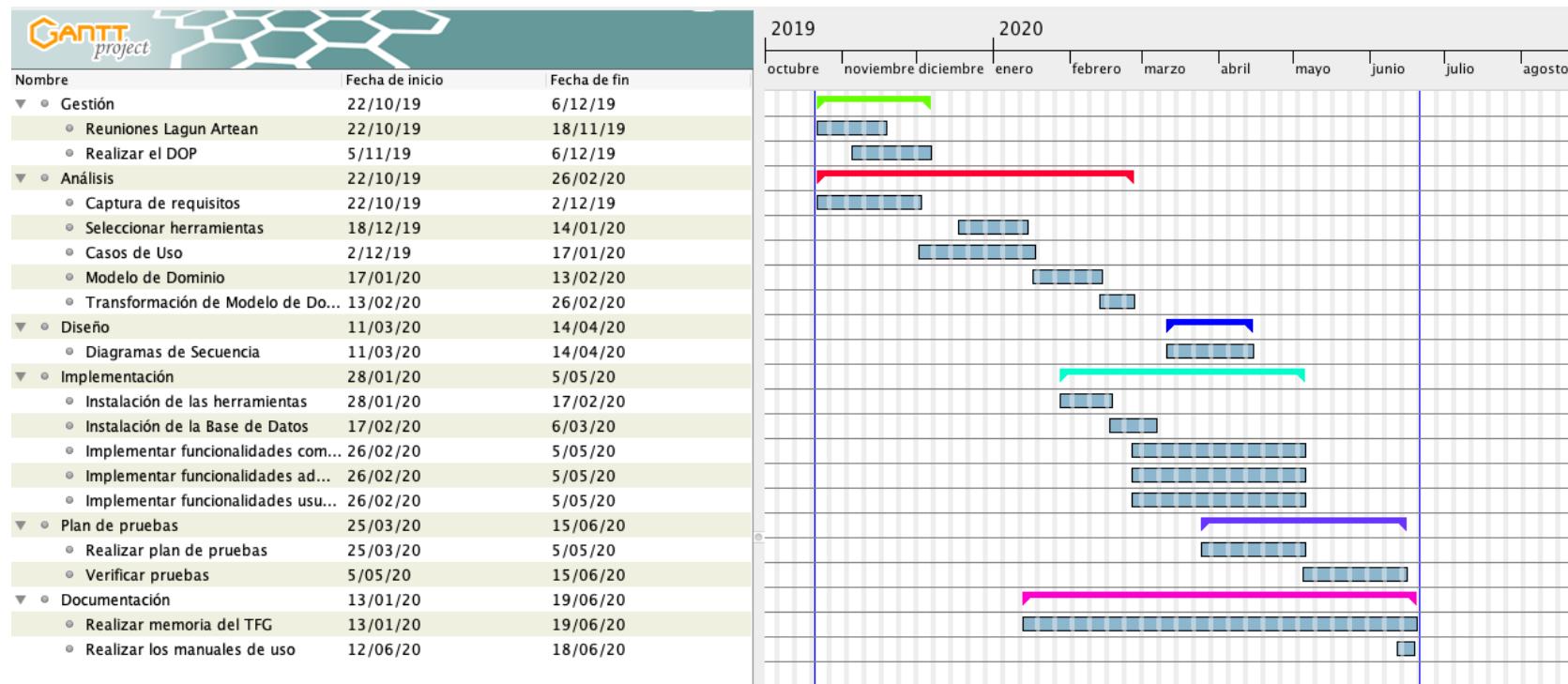


Figura 23: Diagrama de Gantt



2.7. Análisis de alternativas

2.7.1. Alojamiento de la aplicación web

A la hora de elegir el almacenamiento de la aplicación web, se tuvieron distintas opciones en cuenta además de Microsoft Azure y el servidor Apache. Entre las opciones que se tuvieron en cuenta para alojar la base de datos de la aplicación estuvieron Digital Ocean y Node.js + Express. Pero por diferentes motivos que a continuación se explicaran, se rechazaron.

- **Digital Ocean**



Figura 24: Digital Ocean

Al igual que Microsoft Azure, Digital Ocean es una plataforma para la creación de máquinas virtuales privadas. A diferencia de Microsoft Azure, Digital Ocean, Microsoft Azure es más escalable que Digital Ocean, permite SAP en Azure, Oracle en Azure etc. En cambio Digital Ocean, es más económico a la hora de alojar máquinas virtuales.

- **Node + Express**



Figura 25: Node.js + Express

Node.js es un entorno de ejecución de código abierto para la capa de servidor. Usa el lenguaje de programación ECMAScript, el cual está basado en el lenguaje JavaScript. Express es un framework de Node.js que proporciona funcionalidades como el enrutamiento, gestión de sesiones y cookies. Se ha descartado el uso de Node.js + Express para nuestra aplicación, ya que esta iba a ser alojada en un servidor Apache.



2.7.2. Bases de Datos

A la hora de elegir el almacenamiento para la base de datos, se tuvieron distintas opciones en cuenta además de MySQL. Entre las opciones que se tuvieron en cuenta para alojar la base de datos de la aplicación estuvieron MongoDB y Firebase. Pero por diferentes motivos que a continuación se explicaran, se rechazaron.

- **MongoDB**



Figura 26: MongoDB

MongoDB es un sistema de bases de datos NoSQL, donde los datos se almacenan mediante documentos BSON, un documento que es una representación binaria de un archivo JSON y además es de código abierto. Una de las mayores diferencias respecto a una base de datos relacional es que no es necesario seguir un esquema para almacenar los datos, ya que una colección, lo que viene siendo una tabla en una base de datos relacional, puede seguir diferentes esquemas.

Las razones por las que se ha descartado el uso de MongoDB para nuestra aplicación es sencillamente porque no se pueden hacer relaciones entre datos, es decir, no se pueden realizar operaciones JOIN. Para poder consultar datos relacionados en dos o más colecciones, tenemos que hacer más de una consulta, lo cual, no es muy eficiente. Por ello, es mejor optar por un sistema de gestión de bases de datos clásico, de esta forma, las operaciones son mas eficientes y es mas fácilmente escalable.



- **Firebase**



Figura 27: Firebase

Firebase es un conjunto de aplicaciones desarrollado por Google desde el año 2014 para la creación de aplicaciones web y móviles. Firebase dispone de dos aplicaciones para la creación de bases de datos: Firebase Firestore y Realtime Database. Estas dos funcionan de una forma muy parecida a MongoDB, ya que se tratan de bases de datos no relacionales, del formato NoSQL, en donde los datos se almacenan en formato JSON.

Este tipo de base de datos no era muy conveniente para usar en nuestro proyecto, ya que necesitábamos realizar relaciones entre las distintas entidades como Ducha y Usuario. Usar este tipo de base de datos, hubiese complicado la implementación de la aplicación y no hubiese sido tan fácil que la aplicación fuese escalable para futuras versiones. Además, este tipo de bases de datos, al de un tiempo tienen un coste si se sobrepasa el numero de lecturas y escrituras que proporcionan gratuitamente.



2.8. Evaluación económica

A la hora de realizar un proyecto, es importante ver el impacto económico que este supondrá y estimar lo que va a costar su desarrollo, así como la amortización del mismo.

Para calcular el coste total del desarrollo de la aplicación, se calculará en diferentes apartados. Primero se calculará los gastos como el software y el hardware. Despues, otros gastos, como pueden ser el gasto de luz o la conexión a internet y por último el gasto de la mano de obra en función de las horas totales invertidas en el proyecto.

2.8.1. Hardware

Respecto al hardware, para desarrollar el proyecto se ha utilizado un MacBook Pro Retina de 13" del año 2014, cuyo precio inicial fue de 1.256,00€. En la siguiente tabla, se muestra el precio y la vida útil, para así, despues poder calcular la amortización mensual del equipo respecto a su vida útil y posteriormente calcular la amortización durante el proyecto.

Equipo	Vida útil estimada	Precio del equipo
MacBook Pro Retina de 13"	96 meses	1.256,00€

Tabla 19: Precio y vida útil del hardware

$$\text{Amortización mensual} = \text{Precio} / \text{Vida útil estimada}$$

$$\text{Amortización durante el proyecto} = \text{Amortización mensual} * \text{Duración del proyecto}$$

Tras haber realizado los diferentes cálculos, la amortización mensual y la amortización durante el proyecto es la siguiente:

- **Amortización mensual:** 13,08€/mes
- **Amortización durante el proyecto:** 117,72€

Además, para este proyecto se ha utilizado una máquina virtual de Microsoft Azure, para el posterior alojamiento de la página web. La máquina virtual que se ha elegido ha sido, una máquina del tipo A1 con 1 CPU, 2GB de RAM, 10 GB de Almacenamiento temporal con sistema operativo Ubuntu 18.04 LTS.

El coste estimado por mes de la máquina virtual es de 25,24€/mes. El coste total de la maquina virtual durante el proyecto es de 227,16€. El coste total del hardware asciende a 344,28€.

2.8.2. Software

Por otra parte, la mayor parte del software utilizado en la realización de este proyecto o es de código abierto o de uso gratuito. El sistema operativo del ordenador es macOS, por lo que el precio de su licencia ya viene incluida en el



precio del equipo. El editor de LaTeX con el que se esta escribiendo la memoria es Overleaf, de uso gratuito. Por otra parte, el IDE que se esta utilizando para la implementación de la pagina web, IntelliJ IDEA, es de pago, su precio es de 149€/el primer año, y una facturación mensual de 14,90€/mes.

Por lo tanto, el coste total del Software es de 134,10€.

2.8.3. Mano de obra

En cuanto a personal, teniendo en cuenta en base al BOE-A-2019-14977 del 18 de octubre del 2019 sobre el convenio colectivo del sector de Empresas de Ingeniería y Oficinas de Estudios Técnicos, el salario mínimo anual de un analista es de 23.973,88€. El salario se divide en 14 pagas, donde cada paga es de 1.712,42€/mes. Sabiendo que una jornada laboral en una semana son 40 horas y mensualmente son 160 horas, el precio por hora de un trabajador es de 10,70€/hora. Si a todo esto, lo multiplicamos por las 544 horas del proyecto, obtenemos que la mano de obra son 5.820,80€.

2.8.4. Otros gastos

■ Internet

Para poder realizar el proyecto y dado que estamos utilizando servicios en la nube, es necesaria una conexión a Internet. La conexión a Internet utilizada es de fibra óptica simétrica que además incluye linea de teléfono móvil. El precio mensual de la conexión a internet asciende a 87€/mes. El total del gasto de la conexión de internet asciende a 783€.

■ Luz eléctrica

Para calcular el precio de la luz eléctrica, se ha procedido a calcularlo mediante la potencia del adaptador de corriente del equipo utilizado. La potencia del adaptador de corriente es de 60W y la duración del proyecto ha sido durante 9 meses, se han consumido 54 kWh y teniendo en cuenta que el precio de la energía con impuestos incluidos es de 0.1349€/kWh sin discriminación horaria. El total del coste eléctrico es de 7,28€.

2.8.5. Gastos totales

Concepto	Coste
Hardware	344,28€
Software	134,10€
Internet	783€
Luz eléctrica	7,28€
Mano de obra	5.820,80€
Total	7.089,46€

Tabla 20: Gastos totales del proyecto



2.9. Gestión de riesgos

En este apartado se recogen todos los posibles riesgos y problemas que pueden surgir a lo largo de todo el proyecto. También, se recogen posibles soluciones y planes de prevención que podemos realizar en caso de que se produzca algún fallo. Por otro lado, también se recogen las probabilidades de que suceda el fallo así como el impacto que tendría en el proyecto. Para cuantificar la probabilidad y el impacto de cada riesgo se evaluaran de la siguiente manera.

1. Probabilidad

- 1.1. Baja($<30\%$)
- 1.2. Media ($30 < \% \leq 70$)
- 1.3. Alta ($>70\%$)

2. Impacto

- 2.1. Bajo
- 2.2. Medio
- 2.3. Alto

2.9.1. Enfermedad

Descripción	Aparición de una enfermedad que no permita continuar con el proyecto
Prevención	Llevar el proyecto al día o adelantado para superar cualquier contingencia
Plan de contingencia	Invertir mas horas en el proyecto.
Probabilidad	Baja
Impacto	Medio

Tabla 21: Riesgo 1: Enfermedad

2.9.2. Caída de Overleaf

Descripción	Por diferentes problemas técnicos ajenos, Overleaf se cae y no podemos acceder a la memoria.
Prevención	Realizar copias de seguridad de la memoria del Trabajo de Fin de Grado
Plan de contingencia	Traspasar la memoria a otro editor de LaTEX
Probabilidad	Baja
Impacto	Medio

Tabla 22: Riesgo 2: Caída de Overleaf



2.9.3. Caída de Microsoft Azure

Descripción	Microsoft puede tener caídas en sus servidores y por lo tanto tener caídas en nuestro servidor
Prevención	Tener una instancia alternativa en otra nube
Plan de contingencia	Alojar la web en otro servicio en la nube
Probabilidad	Baja
Impacto	Alto

Tabla 23: Riesgo 3: Caída de Microsoft Azure

2.9.4. Ataques cibernéticos

Descripción	Como nuestra aplicación esta alojada en los servidores de Microsoft, puede sufrir ataques cibernéticos
Prevención	Realizar copias de seguridad de nuestro alojamiento
Plan de contingencia	Alojar la web en otro servicio en la nube o restaurar la copia de seguridad
Probabilidad	Baja
Impacto	Alto

Tabla 24: Riesgo 4: Ataques cibernéticos

2.9.5. Mal diseño del proyecto

Descripción	Es posible realizar un mal diseño inicial si no se tienen buenos conocimientos sobre análisis y diseño
Prevención	La realización del análisis y diseño de la aplicación debe ser revisada mas de una vez
Plan de contingencia	No realizar cambios rápidos de diseño durante o antes de la implementación
Probabilidad	Baja
Impacto	Alta

Tabla 25: Riesgo 5: Mal diseño del proyecto



2.9.6. Falta de tiempo para realizar pruebas

Descripción	Por falta de tiempo no disponer de tiempo para realizar pruebas
Prevención	Cumplir con la planificación
Plan de contingencia	Atrasar tareas menos importantes
Probabilidad	Baja
Impacto	Medio

Tabla 26: Riesgo 6: Falta de tiempo para realizar pruebas

2.9.7. Rotura del equipo de trabajo

Descripción	El equipo de trabajo se averíe durante la realización del proyecto
Prevención	Llevar el mantenimiento correcto del equipo
Plan de contingencia	Reparar el equipo o sustituirlo por otro
Probabilidad	Baja
Impacto	Baja

Tabla 27: Riesgo 7: Rotura del equipo de trabajo

2.9.8. No cumplir con la LOPD/RGPD

Descripción	A la hora de tratar datos personales el no cumplimiento de la legislación vigente
Prevención	Lectura y cumplimiento de la LOPD/RGPD
Plan de contingencia	Realizar las acciones impertinentes por incumplimiento de las leyes
Probabilidad	Baja
Impacto	Medio

Tabla 28: Riesgo 8: No cumplir con la LOPD/RGPD



2.9.9. El cliente abandone el proyecto

Descripción	El cliente decide abandonar el proyecto por algún motivo
Prevención	Cumplir con la planificación y mostrarle al cliente un proyecto correcto y que se adecue a sus requisitos.
Plan de contingencia	Convencer al cliente de que no abandone el proyecto
Probabilidad	Baja
Impacto	Alto

Tabla 29: Riesgo 9: El cliente abandone el proyecto

2.9.10. Confinamiento por el COVID-19

Descripción	Debido a confinamiento por el COVID-19 o mayormente conocido por el Coronavirus, existe un posible riesgo de confinamiento
Prevención	Llevar el proyecto al día o adelantado para superar cualquier contingencia
Plan de contingencia	Invertir mas horas en el proyecto y comunicación con el cliente vía telemática
Probabilidad	Alta
Impacto	Medio

Tabla 30: Riesgo 10: Cuarentena o confinamiento por el COVID-19



3. Captura de requisitos

En este apartado se hablará de la captura de requisitos de la aplicación. Primero hablaremos de las reuniones que se tuvieron con los administradores de Lagun Artean para obtener los requisitos del sistema y de los distintos métodos que usamos para obtener la mayor cantidad de ideas posibles.

Tras ello, hablaremos del modelo de casos de uso, donde se reflejan las distintas funcionalidades del sistema, junto a su jerarquía de actores, en la cual se describirá todos los actores presentes en la aplicación. Posteriormente hablaremos del modelo de dominio de la aplicación, donde se hablará de las distintas entidades y relaciones de la aplicación.

3.1. Reuniones en Lagun Artean

Al ser un proyecto real, antes de comenzar con los Casos de Uso, nos reunimos con los administradores del servicio de atención diurna IREKIA. Estas reuniones sirvieron para conocer como funciona actualmente el sistema y poder ver los problemas existentes y así poder ofrecer una solución. Para tener unas actas de reunión de las reuniones, se procedió a crear un cuaderno de bitácora (Figura 28), que servirá de memoria de las reuniones.

28/10/2019

Asisten: Joseba, Bego, Amaia, Patricia, Borja, Ander M., Urko y Ander D.

Patricia y Amaia nos cuentan con un poco más de profundidad los problemas a los que se enfrentan diariamente, tanto ellas como las personas usuarias de los servicios de lagun artean. La reunión la terminamos visitando las instalaciones.

Dada la cantidad de servicios y de problemáticas, a veces específicas y a veces interrelacionadas que se observan, se decide organizar una reunión de trabajo en la que se investigue de forma colaborativa en la definición de los términos de referencia.

En esta reunión participarán también personas usuarias y técnicas de lagun artean (Amaia y Patricia se encargan de seleccionarlas).

Bego nos presenta la que, a priori, se perfila como herramienta a utilizar (odoo). Se decide posponer una explicación más exhaustiva de la herramienta hasta después de la identificación.

Tareas asignadas:

- Añadir a las dos personas que faltaban en el chat de whatsapp - Joseba
- Crear y compartir la carpeta de drive - Joseba
- Dejar en el drive algo de información de la dinámica - Joseba
- Abrir un doodle para el Taller de Identificación Colaborativa - Joseba
- Seleccionar a más personas para este taller - Patricia y Amaia

Figura 28: Árbol de problemas de las reuniones

La primera reunión que se tuvo con los administradores del sistema, fue mas de toma de contacto para conocer como funcionan generalmente y poder ver sus instalaciones y así hacernos una idea de como son sus instalaciones. Después de esta primera toma de contacto, se procedió a quedar para una siguiente reunión para ver como funciona el servicio ya en acción. En esta segunda reunión se



pudo observar la falta de recursos que tienen, ya que en ese día las duchas no estaban disponibles ya que tenían problemas de humedades y las tuvieron que cerrar, por lo que los usuarios no pudieron usarlas. Además, se pudo observar la cola que había en la entrada del local para poder acceder ya que era un día de lluvia y se había formado cola con bastante tiempo de antelación antes de que se iniciase el servicio.

En las siguientes reuniones se utilizaron diversas técnicas para la obtención de ideas de las posibles soluciones que se podrían dar al problema. Entre ellas esta la lluvia de ideas o *Brainstorming* y el árbol de problemas, ambas técnicas se usaron conjuntamente para ordenar la solución de los distintos problemas existentes.

- **Lluvia de ideas:** Para poder construir el árbol de problemas, se organiza una lluvia de ideas, para intentar abordar los problemas existentes. A través de esta lluvia de ideas, se pretende identificar todos los problemas existentes para así después hallar la solución a cada uno de los problemas.
- **Árbol de problemas:** El árbol de problemas (Figura 29), es un diagrama de causa-efecto. Muchas veces un mismo problema, genera otros problemas, como es el caso de las duchas. El problema es la falta de recursos, lo que genera grandes colas y a su vez un impacto acústico y visual junto a los problemas de los vecinos.

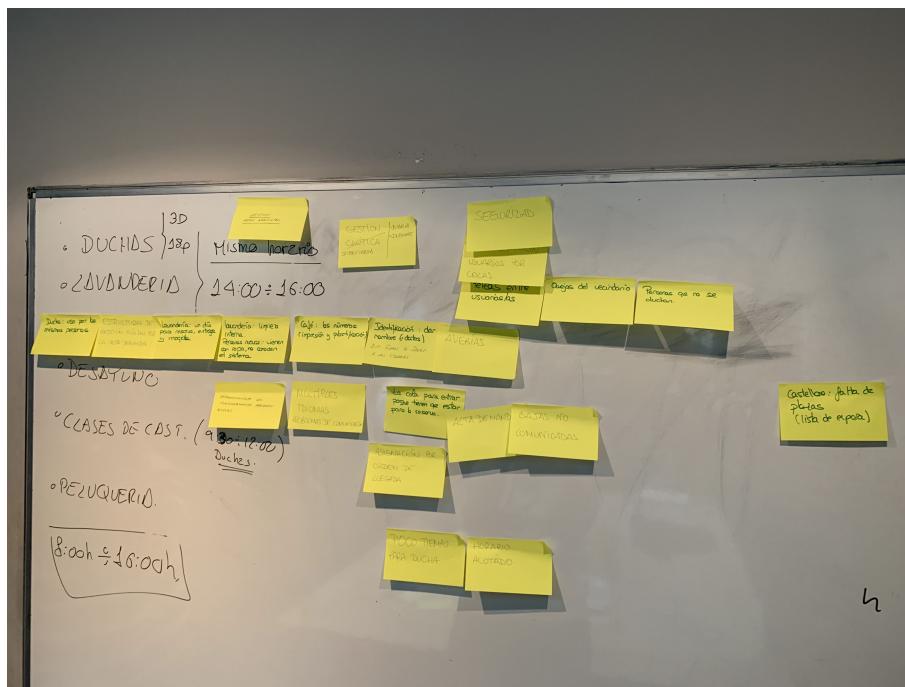


Figura 29: Árbol de problemas de las reuniones

En la última reunión, se presentó a los administradores del servicio de atención diurna, un prototipo (Figura 30) para que pudiesen interactuar con él.



Tras presentárselo, se les hizo un cuestionario sobre la experiencia de uso del prototipo y se recogieron sus opiniones para así aplicar las mejoras convenientes.

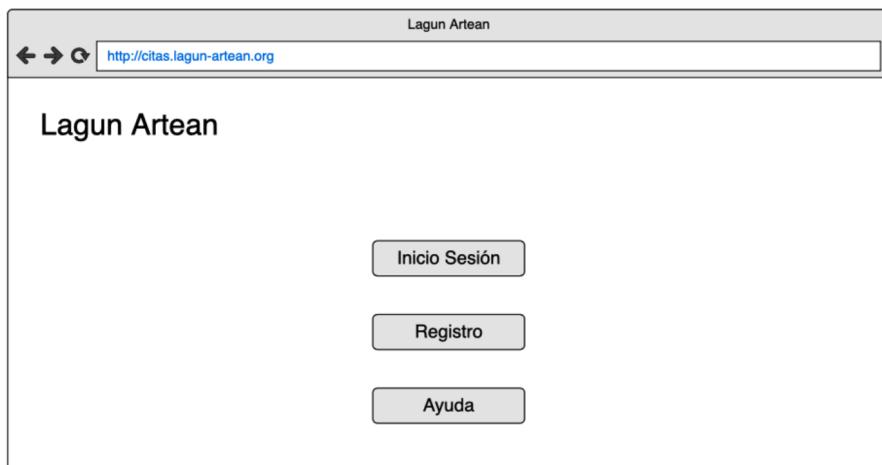


Figura 30: Prototipo Página Web



3.2. Casos de Uso

A continuación se presenta el diagrama de Casos de Uso. En el diagrama se han definido 3 actores. Cada uno tiene asignado varias funcionalidades que se explican en el Anexo A.

3.2.1. Casos de Uso para cualquier usuario

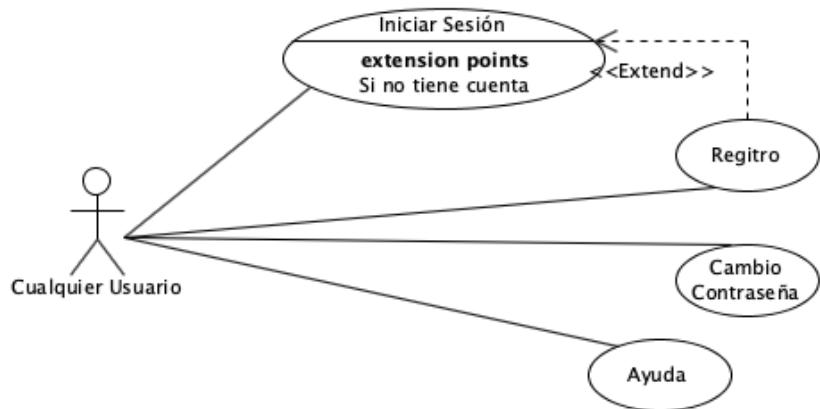


Figura 31: Diagrama de Casos de Uso para cualquier usuario

3.2.2. Casos de Uso para los usuarios

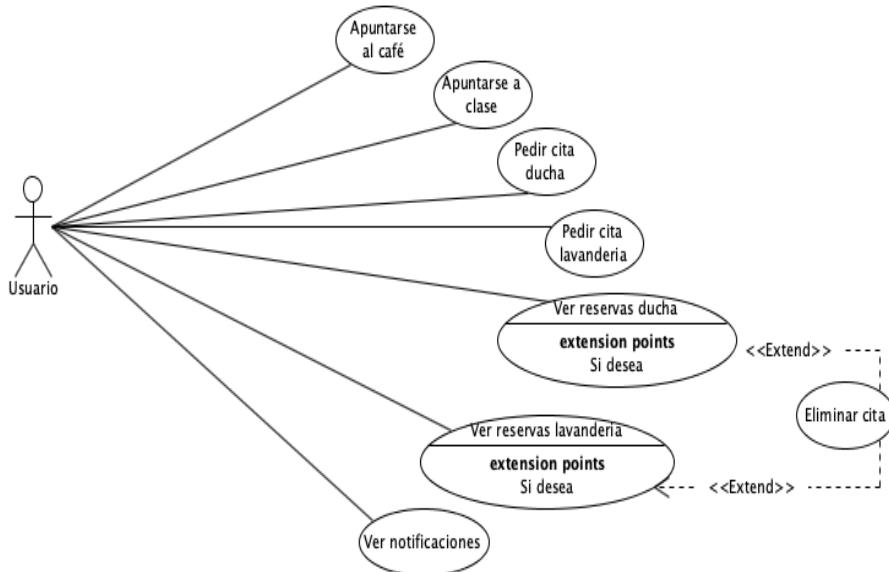


Figura 32: Diagrama de Casos de Uso para los usuarios



3.2.3. Casos de Uso para los administradores

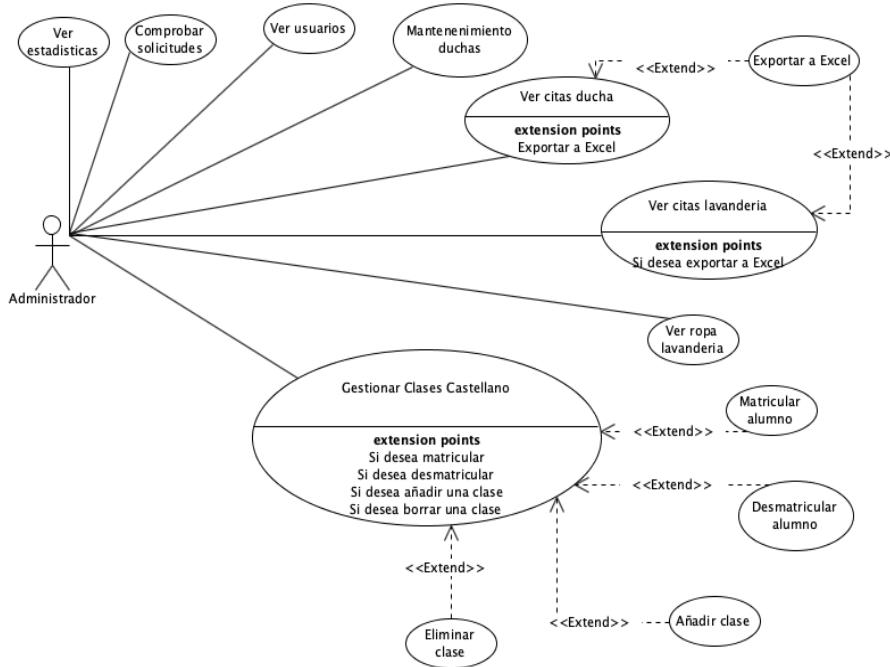


Figura 33: Diagrama de Casos de Uso para los administradores

3.2.4. Jerarquía de Actores

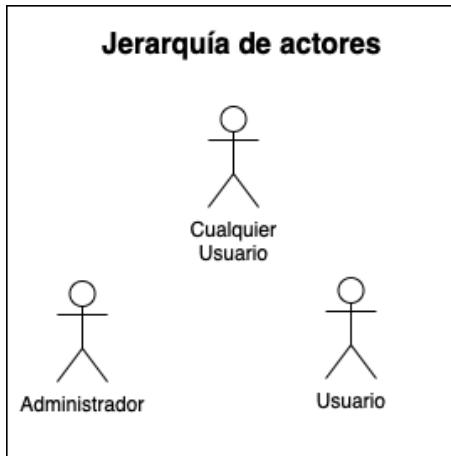


Figura 34: Jerarquía de Actores

Se han definido 3 actores, los cuales se van a explicar a continuación:

- **Cualquier usuario:** El actor cualquier usuario, se refiere al actor que hace funcionalidades comunes, como puede ser iniciar sesión, registrarse,



acceder al menú de ayuda o en el caso de que se le haya olvidado la contraseña, cambiarla.

- **Usuario:** Un usuario, es la persona que hace uso de las instalaciones de Lagun Artean. Un usuario puede realizar las acciones de apuntarse a una clase de castellano, pedir cita para la ducha o pedir cita para la lavandería. Si un usuario ha pedido cita para la ducha o la lavandería, tiene la opción de poder acceder al menú de ver sus citas para la ducha o ver sus citas para la lavandería.
- **Administrador:** El actor administrador, es la persona que gestiona el servicio de atención diurna IREKIA. Cuando un usuario se ha pre registrado en el sistema, el administrador tiene la opción de ver las solicitudes de registro y en tal caso completa el registro de los usuarios.

Cuando hay usuarios en el sistema, este actor puede ver los usuarios del sistema y dentro de esta opción puede o bien editarlos o bien eliminarlos. Además, puede ver las citas para un día concreto para la ducha y para la lavandería. Si un día, una ducha está averiada, este usuario tiene la opción a través del mantenimiento de duchas, poder marcar la ducha averiada para comunicárselo a los usuarios.

Como en el servicio, existe la opción de dar clases de castellano, el administrador es la persona que las gestiona, este actor tiene la función de añadir clases nuevas o eliminar las que había y también la función de matricular nuevos usuarios en las clases.

Para una mejor gestión del servicio, el administrador tiene la opción de poder recoger estadísticas. Desde ese menú el administrador puede recoger distintos tipos de estadísticas.



3.3. Modelo de Dominio

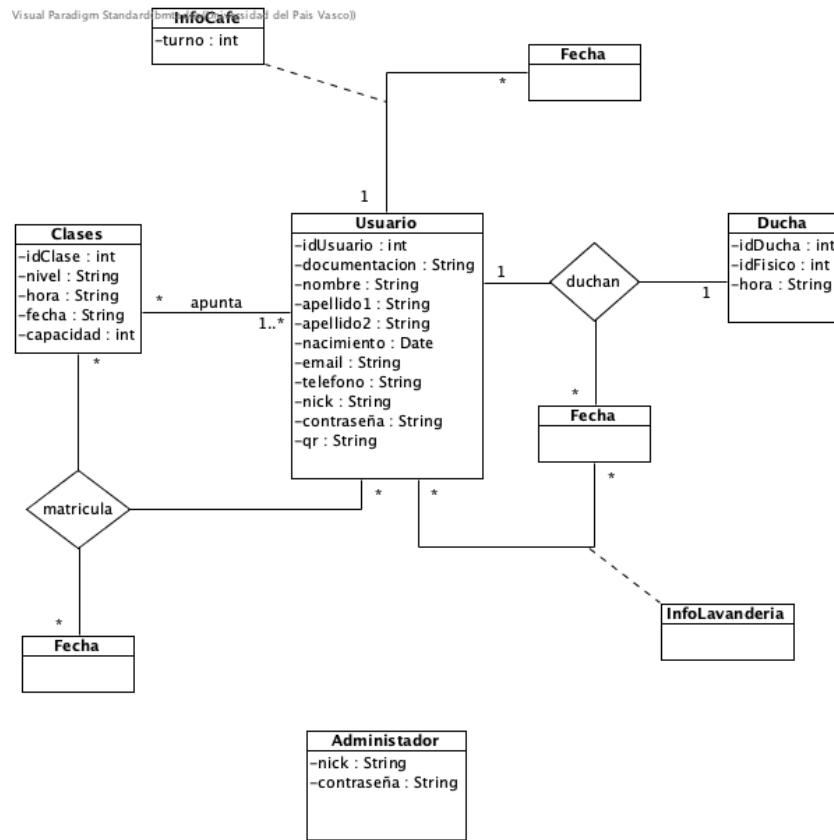


Figura 35: Modelo de Dominio

A continuación se muestra el Modelo de Dominio. En él se describen las distintas entidades, sus atributos y relaciones.

■ Entidades

- **Clases**: La entidad `clases` almacena toda la información necesaria para sobre las clases de castellano.
- **Usuario**: La entidad `usuario` almacena todo lo necesario con la información de los usuarios en el sistema. Esta información es primordial para que el usuario pueda hacer uso de las duchas y de la lavandería.
- **Fecha**: La entidad `fecha` almacena la fecha para los distintos usos que pueden hacer los usuarios del sistema.
- **Administrador**: Esta entidad almacena los datos del administrador. Al no tener relación con ninguna de las demás entidades, se queda separada.
- **Ducha**: Esta entidad, almacena la información sobre las duchas del sistema.



-
- **InfoLavanderia:** Guarda toda la información sobre la lavandería.
 - **InfoCafe:** Guarda toda la información sobre los cafés. A partir de esta información se puede obtener cuantos cafés ha tomado un usuario o cuantos cafés ha habido por día.



■ Relaciones

- **Matricula:** Es la unión entre clase, ducha y usuario. Es cuando un usuario se ha añadido en una fecha a una clase.
- **Apunta:** Cuando un usuario se añade a una clase provisionalmente.
- **Duchan:** Es la unión entre fecha, ducha y usuario. Se almacenan los usuarios que se duchan en una fecha concreta.
- **Lavandería:** Es la unión entre usuario y fecha, en la que se guardan distintos atributos de información para la lavandería.



4. Análisis y Diseño

4.1. Transformación de Modelo de Dominio a Diagrama de Base de Datos

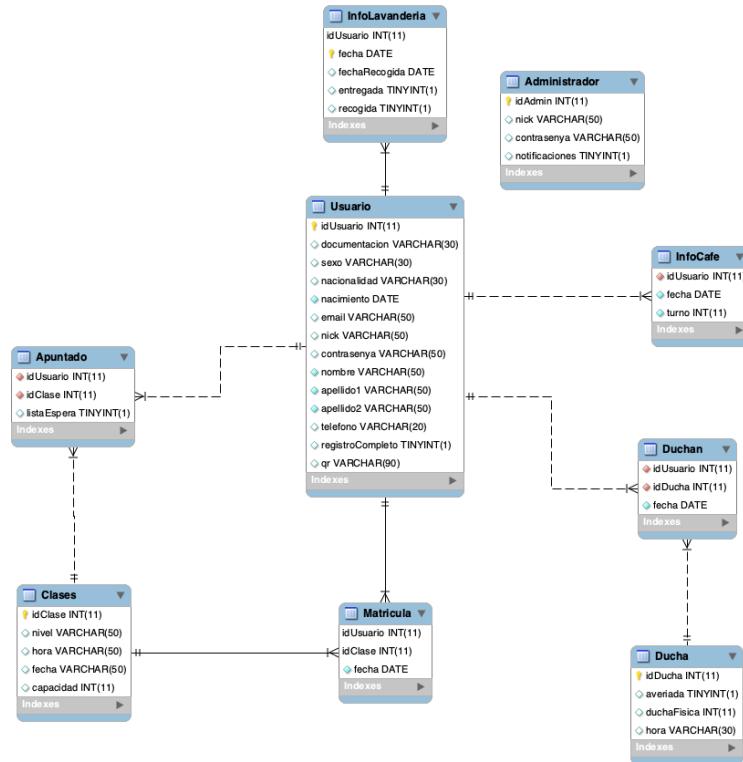


Figura 36: Diagrama de Bases de Datos

Tras haber realizado el Modelo de Dominio, se ha convertido en un diagrama de Base de Datos. Para ello, hemos utilizado las reglas establecidas para hacer el cambio. Para cada entidad del Modelo de Dominio se ha establecido una tabla en la base de datos. En el caso de las relaciones entre varias entidades, se han convertido tablas, en ella estarán las claves primarias a con las tablas que se relacionan.



4.2. Tablas de la Base de Datos

- **Usuario:** En la tabla Usuario, se almacenan todos los datos relacionados con los usuarios. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.

Usuario	
!	idUsuario INT(11)
◊	documentacion VARCHAR(30)
◊	sexo VARCHAR(30)
◊	nacionalidad VARCHAR(30)
◊	nacimiento DATE
◊	email VARCHAR(50)
◊	nick VARCHAR(50)
◊	contrasenya VARCHAR(50)
◊	nombre VARCHAR(50)
◊	apellido1 VARCHAR(50)
◊	apellido2 VARCHAR(50)
◊	telefono VARCHAR(20)
◊	registroCompleto TINYINT(1)
◊	qr VARCHAR(90)
Indexes	

Figura 37: Tabla Usuario

- **idUsuario:** Guarda un id único para la tabla usuario, este valor será un entero que se ira auto incrementando según aumenta la tabla.
- **documentacion:** Guarda la documentación del usuario en caso de que el usuario disponga de ella, para ello se ha decidido que sea una cadena de caracteres.
- **nacimiento:** Guarda la fecha de nacimiento del usuario, de esta forma los administradores, pueden saber cuantos años tienen los usuarios.
- **sexo:** Almacena el sexo de cada usuario, este campo será útil para poder sacar estadísticas de uso del servicio.
- **nacionalidad:** Este campo de la base de datos sirve para almacenar la nacionalidad del usuario, se almacenara una cadena de caracteres.
- **email:** Sirve para almacenar en caso de que disponga el usuario un email, se almacenara una cadena de caracteres.
- **nick:** El nick sirve para iniciar sesión y poder buscar un usuario en la base de datos, el nick es único y es autogenerado por la base de datos, se trata de una cadena de caracteres autogenerada bajo un patrón.



- **contrasenya:** Almacena la contraseña de cada usuario de la base de datos, es una cadena de caracteres y para que no haya problemas con la base de datos, se ha procedido a eliminar la 'ñ' del atributo.
 - **nombre:** Almacena el nombre del usuario en la base de datos, se almacena una cadena de caracteres.
 - **apellido1:** Almacena el primer apellido del usuario en la base de datos, se almacena una cadena de caracteres.
 - **apellido2:** Almacena el segundo apellido del usuario en la base de datos, se almacena una cadena de caracteres.
 - **telefono:** En caso de que el usuario disponga de teléfono, se almacena como cadena de caracteres su teléfono.
 - **registroCompleto:** Para distinguir a los usuarios que ha realizado ya el registro completo, se ha añadido una variable booleana que indica si el registro ha sido completado o no.
 - **qr:** Para facilitar a los administradores la búsqueda de usuarios, se ha añadido un código QR, que se almacena como una cadena de caracteres de la URL de donde se genera.
- **Administrador:** En la tabla Administrador, se almacenan todos los datos relacionados con el administrador. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.

Administrador	
idAdmin	INT(11)
nick	VARCHAR(50)
contrasenya	VARCHAR(50)
notificaciones	TINYINT(1)
Indexes	

Figura 38: Tabla Administrador

- **idAdmin:** Guarda un id único para la tabla administrador.
- **nick:** Guarda el nick del administrador en la base de datos, para posteriormente poder iniciar sesión.
- **contrasenya:** Guarda la contraseña del administrador en la base de datos, se guarda como una cadena de caracteres.
- **notificaciones:** En este atributo se guarda una variable booleana para el caso en el que el administrador quiera recibir notificaciones cuando un usuario se registra, un usuarios pide una ducha... De esta forma, si la variable esta a false, el administrador no recibirá ninguna notificación.



-
- **Clases:** En la tabla Clases, se almacenan todos los datos relacionados con las clases de castellano. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.



Figura 39: Tabla Clases

- **idClase:** Guarda un id único para la tabla clase.
- **nivel:** En este atributo se almacena el nivel que tiene asignada cada clase, en el sistema, los administradores elegirán que nivel tendrá cada clase.
- **hora:** Se guarda la franja horaria en la que se imparte la clase de castellano, al ser una franja horaria lo mejor para almacenarla es una cadena de caracteres.
- **fecha:** Se almacena los días de la semana que se da la clase, una clase se puede dar en dos días de la semana distintos, para almacenar de una mejor forma las clases, se almacenan como una cadena de caracteres.
- **capacidad:** Se almacena la capacidad de cada clase de castellano.



-
- **Apuntado:** En la tabla Apuntado, se almacenan todos los datos relacionados con los usuarios que se apuntan a clases de castellano pero que aun no se han matriculado definitivamente. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.



Figura 40: Tabla Apuntado

- **idUsuario:** Se almacena el id del usuario que se ha apuntado a una clase.
 - **idClase:** Se almacena el id de la clase, de la clase que se ha apuntado el usuario.
 - **listaEspera:** Se almacena un booleano si ese usuario esta esperando a apuntarse a esa clase cuando se liberen nuevas plazas.
- **Matricula:** En la tabla Matricula, se almacenan todos los datos relacionados con los usuarios que se han matriculado en clases de castellano. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.

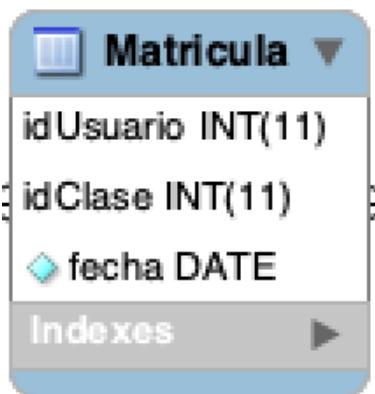


Figura 41: Tabla Matricula

- **idUsuario:** Se almacena el id del usuario que se ha apuntado a una clase.
- **idClase:** Se almacena el id de la clase, de la clase que se ha apuntado el usuario.



- **fecha:** Se almacena la fecha de los días de la semana que el usuario acudirá a las clases para las que se ha matriculado.

- **Ducha:** En la tabla Ducha, se almacenan todos los datos relacionados con las duchas. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.

Ducha	
!	idDucha INT(11)
◇	averiada TINYINT(1)
◇	duchaFisica INT(11)
◇	hora VARCHAR(30)
Indexes	

Figura 42: Tabla Ducha

- **idDucha:** Se almacena el id de la ducha de la tabla, como en total cada día hay 18 duchas de 15 minutos cada una, habrá 18 id de ducha.
- **duchaFisica:** Al haber únicamente 3 duchas en el local de Lagun Artean, guardamos el id físico de cada una de ellas. Como además, hay un turno cada 15 minutos para cada ducha, las 3 duchas tienen turno en la misma franja horaria. Además, es mas útil para los usuarios hacer cola para la ducha que tienen asignada.
- **hora:** Cada ducha, tiene una franja horaria asignada, para almacenar de la mejor forma las franjas horarias, se procede a almacenar cada franja horaria con una cadena de caracteres.
- **averiada:** Ya que es posible que una ducha este averiada, se procede a almacenar en una variable booleana si una ducha esta averiada o no.



-
- **Duchan:** En la tabla Duchan, se almacenan todos los datos relacionados con los usuarios que se duchan en las duchas. Sirve para relacionar las tablas de Ducha y Usuario. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.

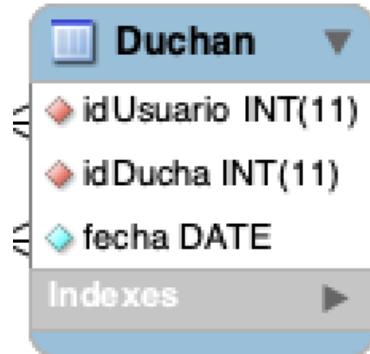


Figura 43: Tabla Duchan

- **idUsuario:** Se almacena el id del usuario que se ha apuntado a una ducha.
- **idDucha:** Se almacena el id de la ducha que se ha apuntado un usuario.
- **fecha:** Se almacena la fecha cuando un usuario se ha apuntado para ducharse en una ducha, de esta forma un usuario se puede duchar en varias duchas en distintas fechas.



-
- **InfoLavanderia:** En la tabla InfoLavanderia, se almacenan todos los datos relacionados con la lavandería de Lagun Artean. A continuación, se explican los diferentes atributos que se almacenan en esta tabla de la base de datos.

The screenshot shows the MySQL Workbench interface with the 'InfoLavanderia' table selected. The table has the following structure:

	idUsuario INT(11)
	fecha DATE
	fechaRecogida DATE
	entregada TINYINT(1)
	recogida TINYINT(1)
	Indexes

Figura 44: Tabla InfoLavanderia

- **idUsuario:** Se almacena el id del usuario que se ha apuntado a la lavandería.
- **fecha:** Se almacena la fecha de cuando un usuario tiene la cita con la lavandería para entregar la ropa.
- **fechaRecogida:** En este campo, se almacena la fecha que se le indica al usuario para que pase a recoger la ropa que ya está lavada.
- **recogida:** Este campo, es un campo booleano que le indica al usuario si la ropa ha sido ya entregada en la lavandería o no.
- **entregada:** Cuando a un usuario ya se le ha entregado la ropa lavada, se marca en la base de datos que la ropa ya ha sido entregada a su usuario.



-
- **InfoCafe:** En la tabla InfoCafe, se almacenan todos los datos relacionados con la gestión del café de Lagun Artean. En ella, se almacenarán los números del café para cada día que haya servicio del café.



Figura 45: Tabla InfoCafe

- **idUsuario:** Se almacena el id del usuario que se ha apuntado a una ducha.
- **fecha:** Se almacena la fecha de cuando un usuario tiene la cita con la lavandería para entregar la ropa.
- **turno:** En este campo, se almacena el turno de cada usuario que tiene cada dia.



4.3. Diagrama de clases para la aplicación de escritorio

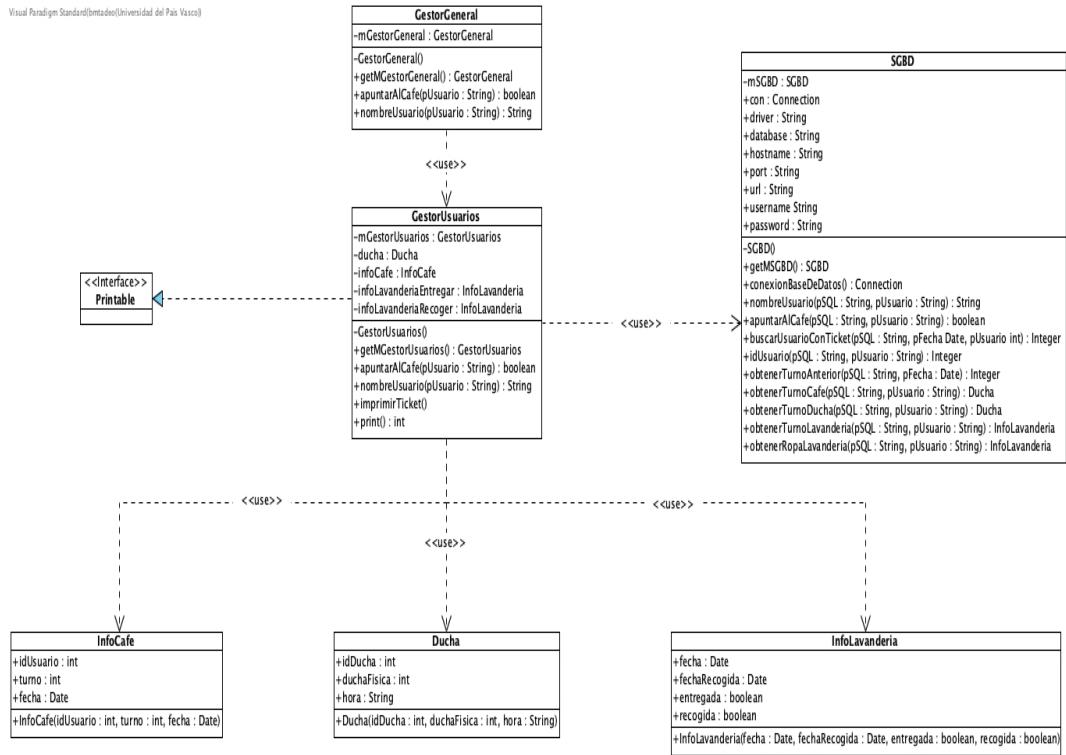


Figura 46: Diagrama de Bases de Datos

A continuacion, se explicara las clases del diagrama.

- **GestorGeneral:** Es la fachada del sistema, usaremos esta clase para poder acceder a otros gestores en el caso de que los haya, por lo que hace la aplicación mas ampliable.
- **GestorUsuarios:** Esta clase se encarga de obtener todos los datos relacionados con los usuarios, apuntarse al café, obtener sus citas...
- **SCBD:** Es la clase encargada para realizar las consultas con la base de datos.
- **Ducha:** Representa lo que es una ducha en el sistema.
- **InfoCafe:** Representa la informacion sobre el servicio del café en el sistema.
- **InfoLavanderia:** Representa la informacion sobre la lavandería en el sistema.



4.4. Códigos QR

Los códigos QR fueron creados por Denso Wave subsidiaria japonesa en el Grupo Toyota en el año 1994. Actualmente, los códigos QR son de uso libre. El objetivo de los creadores era que el código se tuviera que leer a alta velocidad.

Un código QR es un código de barras bidimensional que almacena datos, desde textos simples, a enlaces a páginas web, correos electrónicos... En nuestro caso, estos códigos QR serán utilizados para almacenar el nick del usuario que visita las instalaciones de Lagun Artean que va a utilizar alguno de los servicios. Cuando un usuario se registra en el sistema se le proporciona un nick y este va asignado a un código QR. El usuario podrá visualizarlo en una aplicación móvil o bien, cuando se completa el registro del usuario en las instalaciones de Lagun Artean, se le puede imprimir su usuario junto a su código QR.

Cuando un usuario accede a las instalaciones de Lagun Artean para hacer uso de los servicios de café, ducha o lavandería, escaneara su código QR que o bien lo tendrá en una aplicación móvil o impreso en una tarjeta. Tras escanear el código QR, el sistema reconocerá que usuario es, le imprimirá un ticket con un turno para el café para ese día, un recordatorio al hora que tenía reservada la ducha y el recordatorio de la cita de lavandería. Si el usuario ya tenía ropa en la lavandería para recoger, se le avisara de que puede pasar ya a recogerla.



Figura 47: Ejemplo de código QR



5. Desarrollo

En este capítulo de la memoria del proyecto, se explicara las diferentes fases del desarrollo de la aplicación. Primero, hablaremos sobre la parte del diseño de la web y de la aplicación para la lectura de códigos QR del servicio del café, además, se hablará de las tecnologías utilizadas durante la implementación.

Por otra parte, se realizara una comparación de las bases de datos que se podrían utilizar en la base de datos o las diferentes opciones para alojar la aplicación web. En el caso de la aplicación de escritorio para leer el código QR de los usuarios cuando usan el servicio de café, se hablará de la API utilizada y de la librería utilizada, para la lectura de los códigos QR.

Para finalizar, al ser un proyecto en el que se guardan datos sobre usuarios reales, es conveniente realizar una mención sobre el uso de la Ley Orgánica de Protección de Datos (LOPD). Por ultimo, se hablaran de los problemas que se han encontrado durante el desarrollo y las soluciones que se han encontrado a estos problemas.

5.1. Diseño de la web

En este apartado del desarrollo, hablaremos sobre el back-end y front-end de la aplicación web. Para ello, se definen y explican las tecnologías utilizadas, como se han instalado y como se han integrado integrado.

5.1.1. Back-end

■ Microsoft Azure

Microsoft Azure es una nube de pago por uso propiedad de Microsoft. Usaremos Microsoft Azure para tener un alojamiento para la aplicación web. Se ha elegido este tipo de almacenamiento, ya que aparte de crear máquinas virtuales, tiene multitud de aplicaciones que se pueden conectar entre si, lo que la hace muy escalable.

■ Apache

Apache es un servidor web HTTP de código abierto, aunque llamamos a Apache un servidor web, no es un servidor físico, sino un software que se ejecuta en un servidor. Su trabajo es establecer una conexión entre un servidor y los navegadores de los usuarios que acceden a la aplicación web. Cuando un usuario accede a una zona de la aplicación web, el navegador envía una petición al servidor web y Apache le devuelve la respuesta con los elementos que le ha solicitado (texto, imágenes,...).

Una de las características mas útiles de Apache para nuestra aplicación, es la de cargar de manera dinámica, los datos de la base de datos en nuestra aplicación web, de esta forma, se carga dinámicamente la pagina web.

5.1.2. Base de datos

Para la base de datos, se ha elegido MySQL. El cual es un sistema de gestión de bases de datos relacional que utilizaremos en la implementación. Se ha elegido



este tipo de bases de datos, ya que permite relacionar los datos de una manera eficiente. Además, como esta alojada en Azure, nos permitirá conectarnos a la base de datos desde la aplicación de escritorio o desde cualquier futura aplicación que se vaya a desarrollar.

5.1.3. Mapa

Es de vital importancia que una persona que no conoce donde esta situado el Centro de Acogida Lagun Artean sepa donde esta y cuales son sus métodos de contacto. Por ello, se ha añadido en el pie de pagina de la pagina web un mapa con la localización del local además de sus dos formas de contacto, teléfono y correo electrónico.

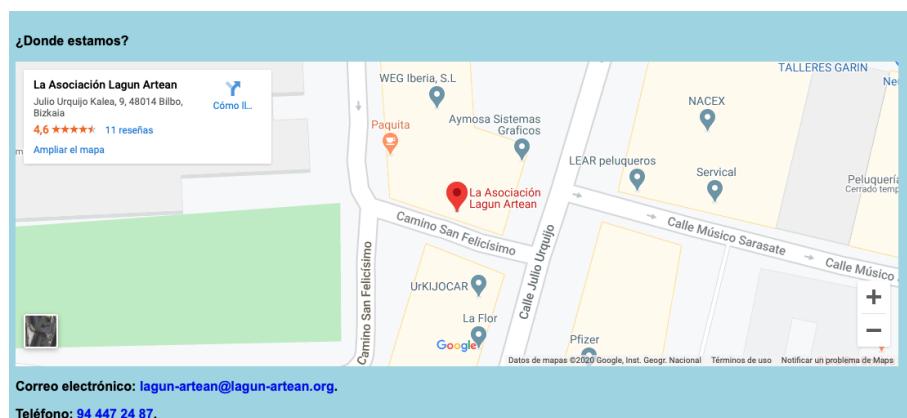


Figura 48: Donde estamos

5.1.4. Soporte multiidioma

Para favorecer la accesibilidad de la aplicación a todos sus usuarios, se le ha dotado a esta la característica de ser multiidioma. En este caso, los idiomas implementados han sido el castellano, el francés y el árabe. Se ha elegido estos tres idiomas ya que son los principales que conocen los usuarios del sistema. Además, como el árabe se lee de derecha a izquierda, se ha adaptado mediante CSS para que se adapte a su modo de lectura.

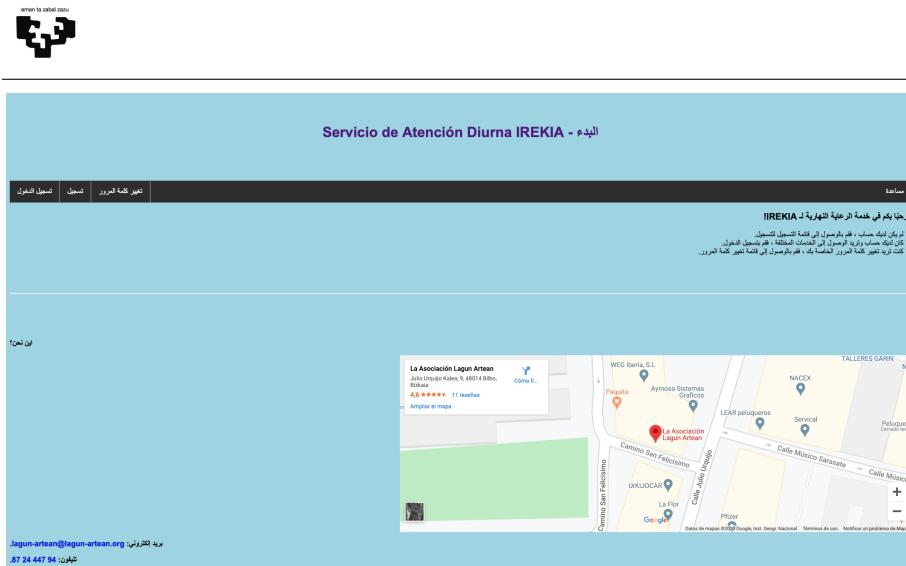


Figura 49: Soporte multiidioma

5.1.5. Soporte multipantalla

Dado que no se trata de una aplicación móvil, se ha pretendido que aun así, sea lo mas adaptable posible para pantallas pequeñas. Por ello, se ha intentado adaptar lo máximo posible a las pantallas de los dispositivos móviles.



Figura 50: Soporte multipantalla



5.2. Diseño de la aplicación

En este capítulo, hablaremos sobre el diseño que se ha llevado para implementar la aplicación de escritorio para los turnos del servicio de café. Para ello, hablaremos de las diferentes API y librerías utilizadas para poder implementar la lectura de códigos QR.

5.2.1. Maven

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>
  <groupId>com.bmtadeo.codigosqr</groupId>
  <artifactId>codigosqr</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>jar</packaging>

  <dependencies>

    <dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
      <version>1.2.17</version>
    </dependency>

    <dependency>
      <groupId>com.github.sarxos</groupId>
      <artifactId>webcam-capture</artifactId>
      <version>0.3.12</version>
    </dependency>

  </dependencies>
</project>
```

Esta herramienta va a ser utilizada para poder instalar las librerías necesarias del proyecto. Las librerías del proyecto se indican como dependencias en el fichero *pom.xml* (Project Object Model) el cual se encarga de recopilar las dependencias necesarias y posteriormente cargarlas en el proyecto. Además de cargar las dependencias necesarias, también muestra el nombre del Artefacto (Artifact ID) y el tipo de empaquetado.

5.3. API Google Charts

Se ha usado la API de Google Charts para reproducir en la página web distintos gráficos que se pueden extraer de los datos de la base de datos del servicio. El funcionamiento de esta mediante peticiones http a una URL, en nuestro caso: <https://developers.google.com/chart/> y con unos determinados parámetros



se nos devolverá una imagen que será la gráfica resultante.

Debido a que este API ha sido desarrollado por Google es gratuito aunque de pago en el que se pasen las peticiones a la URL. Aunque el numero actual de peticiones gratuitas es de 250000, en un futuro en el servidor web se podrían almacenar las gráficas mas utilizadas para así no gastar todas las peticiones.

5.4. API Google Zxing

La API de Google ZXing es una librería de procesamiento de códigos de barra implementada en Java. Gracias a Maven instalaremos esta librería en la aplicación de escritorio para poder leer los códigos QR de los usuarios.

5.5. API Webcam-Capture

Este API se utiliza en combinación con el API de Google Zxing. Esta API sirve para capturar imágenes a partir de la webcam de un ordenador independientemente de si esta integrada o no en el ordenador. La librería integra los drivers necesarios para poder sacar y procesar una imagen independientemente del fabricante y del sistema operativo del ordenador.

5.6. Ley Orgánica de Protección de Datos personales y Garantía de los Derechos Digitales (LOPDGDD)

Debido a que nuestra aplicación almacenara datos sensibles de los usuarios del sistema como pueden ser, documento de identidad si estos tuviesen posesión de el, su contraseña, su fecha de nacimiento, su correo electrónico, sus nombres y apellidos etc. Debemos tener especial cuidado a la hora de almacenar estos tipos de datos.

En el 6 de diciembre de 2018, entro en vigor Ley Orgánica de Protección de Datos personales y Garantía de los Derechos Digitales (LOPDGDD) que adapta al contexto legal español el RGPD europeo que entro en funcionamiento el 25 de mayo de 2018. Esta ley viene a ser una actualización de la Ley Orgánica de Protección de Datos (LOPD) aprobada en 1999 con entrada en vigor en el año 2000.

En esta nueva ley se han visto reforzados los derechos de los usuarios. Por ello, la LOPDGDD, obliga a las personas que gestionan sus datos a:

- Los datos que se recogen de los usuarios solo se deben utilizar para el fin indicado.
- A la hora que un usuario inserta sus datos, se debe especificar que datos se van a recoger y con que fin. También se indicara, a que persona deben acudir para rectificarlos o cancelarlos, ya que el usuario tiene derecho a hacerlo.
- Se debe facilitar un modo de contacto para que el usuario pueda cancelarlos o rectificarlos y quien esta a cargo de los datos.



-
- El usuario debe consentir que proporciona sus datos y saber que su consentimiento puede ser revocado si existe una causa justificada.

Para cumplir con la normativa vigente, se realizaran las siguientes medidas a la hora de la implementación.

- A la hora del registro, se le mostrara al usuario un texto informándole de los datos que se van a recoger y con que fin.
- En el momento de registrarse, se le indica a que persona deben acudir y en que lugar se encuentra, por si desea o bien rectificarlos o bien eliminarlos.

5.7. HTTPS

A diferencia del protocolo HTTP (Hypertext Transfer Protocol), el protocolo HTTPS (Hypertext Transfer Protocol Secure), es un protocolo de red basado en el protocolo HTTP destinado a la transferencia segura de datos, una versión segura del protocolo HTTP. Es esencial el uso de este protocolo para cumplir con la Ley Orgánica de Protección de Datos personales y Garantía de los Derechos Digitales, de esta forma los datos se transfieren de una forma mas segura.

Una sesión segura a diferencia de una sesión no segura se diferencia en que la sesión segura agrega criptografía para codificar la información transmitida, de tal forma que si alguien intercepta la comunicación el contenido no puede ser descifrado sin contar con las llaves correctas. Además, se emplean certificados para garantizar que la comunicación se esta estableciendo con quien se quería mantener la comunicación, así se evita que un intermediario impostor pueda interceptar la comunicación entre los dos individuos, en el caso de que un individuo lo interceptase se produciría un ataque del tipo *Man in the Middle*.

5.8. Lenguajes utilizados

A la hora de implementar tanto la aplicación web como la aplicación de escritorio para el reconocimiento de los códigos QR de los usuarios que acuden a tomar el café, se han utilizado diferentes lenguajes.



5.8.1. Java



Figura 51: Java

Java es un lenguaje de programación orientado a objetos creado en el año 1995. Se ha elegido este lenguaje para realizar la aplicación de lectura de códigos QR para el servicio de café, ya que es seguro, rápido y fiable a la hora de hacer este tipo de aplicaciones. Además, es muy dinámico a la hora de realizar este tipo de aplicaciones que requieren de librerías externas.

5.8.2. HTML 5



Figura 52: HTML (HyperText Markup Language)

El lenguaje HTML (HyperText Markup Language) es un lenguaje de marcas que se utiliza para desarrollar páginas web de Internet. A este tipo de lenguaje de marcas se le puede añadir código de otros lenguajes como PHP o JavaScript para hacer páginas web más dinámicas.



5.8.3. JavaScript

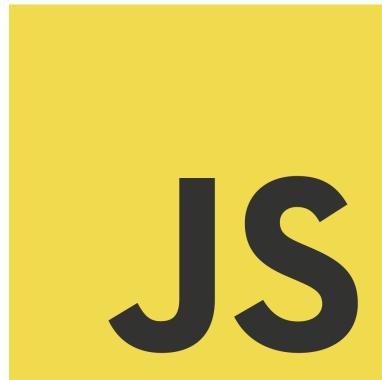


Figura 53: JavaScript

JavaScript es un lenguaje de programación interpretado. Es utilizado en conjunto con HTML o PHP para la realización de páginas web dinámicas. En nuestro caso se ha utilizado JavaScript, para la validación de los datos de los usuarios en la parte de cliente y para el uso de la API de Google Charts.

5.8.4. PHP



Figura 54: PHP

PHP (Hypertext Preprocessor) es un lenguaje de programación del lado del servidor para hacer páginas web más dinámicas. En el desarrollo del proyecto ha sido utilizado la conexión, extracción y modificación de datos de la base de datos, así como, mostrar datos dinámicamente a los usuarios. Además, hemos hablado de que se ha usado JavaScript para la validación de entrada de datos en la parte cliente. Por su parte, PHP se ha utilizado para la validación de datos en el lado del servidor, ya que si no se validasen en el lado del servidor podría ocurrir ataques informáticos contra la base de datos y el sistema en general.

5.9. Problemas y soluciones

A lo largo de todo el desarrollo de la aplicación se han encontrado diversos problemas, los cuales se comentarán a continuación como las soluciones encontradas a estos problemas.



Uno de los principales problemas durante la implementación ha sido cuando se estaba empezando a implementar la aplicación para los turnos de servicio del café. Desde un principio, siempre se ha querido que esta aplicación fuese la mas sencilla y automática de usar, básicamente que fuese parecido a los turnos de cualquier carnicería o pescadería, recoger el turno y esperar a que te llamasen. Para esta aplicación se plantearon dos alternativas o bien unirla a la aplicación web o hacer una aplicación de escritorio aparte que estuviese conectada a la misma base de datos.

Uno de los problemas de desarrollar la aplicación como una aplicación web, era básicamente que cada vez que se imprimiese un ticket iba a salir una ventana en el navegador con las distintas opciones de impresión, por lo que no iba a ser tan automático e iba a ser mas lento todo el proceso de escaneo del código QR y posterior recogida del ticket. Con la solución encontrada de implementar una aplicación aparte, es que una vez escaneado el código QR del usuario, la impresión es automática ya que se indica en el propio código las propiedades de impresión y no es necesaria ninguna interacción del usuario con el sistema, por lo que la hace mas sencilla de utilizar.

En relación al problema anterior, ha existido varios problemas con el tema de las webcam, el equipo con el que se ha desarrollado la aplicación usa macOS como sistema operativo principal. A la hora de hacer las correspondiente pruebas de que el código implementado funcionaba, no dejaba usar la webcam, por lo que se ha tenido que probar en Windows 10 en el mismo dispositivo para comprobar que la aplicación funciona correctamente. Esto se debe a que en macOS las aplicaciones hay que firmarlas, para que el sistema operativo las detecte como confiables.

Otro problema, ha sido el tema que los correos electrónicos que se envían cuando un usuario se había registrado, reservado una cita... no se enviaban porque no estaban cifrados, por lo que estos acababan en la carpeta de spam del destinatario. Para solucionar este error, se procedido a enviarlos de forma segura, para que así, no acabasen en la carpeta de spam del destinatario.



6. Pruebas unitarias

En este apartado se va a hablar sobre las pruebas realizadas durante el desarrollo. Mientras que en otros desarrollos, como la creación de aplicaciones en Java en las que existen herramientas específicas para la realización de este tipo de pruebas, como pueden ser los JUnits. En cambio, para testear el funcionamiento de una página web, la manera más práctica es probar desde el navegador de un usuario.

6.1. Pruebas relativas respecto al registro

Caso	Características	Resultado esperado	Resultado ocurrido	Nota
1a	Registro con datos completos	Redirección a página principal	Hay redirección	Error en la conexión a la BBDD
1b	Registro con datos completos	Redirección a página principal	Hay redirección	Error al insertar en la BBDD
1c	Registro con datos completos	Redirección a página principal	Hay redirección	Correcto
2a	Registro con datos incompletos	Mensaje de error y marcar campos incompletos	Mensaje de error y marca los campos	Correcto

Tabla 31: Pruebas relativas respecto al registro



6.2. Pruebas relativas respecto al inicio y cierre de sesión

Caso	Características	Resultado esperado	Resultado ocurrido	Nota
3a	Inicio sesión con administrador y datos incorrectos	Mensaje de error	Mensaje de error	Correcto
4a	Inicio sesión con usuario y datos incorrectos	Mensaje de error	Mensaje de error	Correcto
5a	Inicio sesión con administrador y datos correctos	Redirección al panel de administrador	Redirección a la pagina principal	Error
5b	Inicio sesión con administrador y datos correctos	Redirección al panel de administrador	Redirección al panel de administrador	Correcto
6a	Inicio sesión con usuario y datos correctos	Redirección a panel de usuario	Hay redirección	Correcto
7a	Inicio sesión con administrador y datos incompletos	Mensaje de error	Mensaje de error	Correcto
8a	Inicio sesión con usuario y datos incompletos	Mensaje de error	Redirección a panel de usuario	Error
8b	Inicio sesión con usuario y datos incompletos	Mensaje de error	Mensaje de error	Correcto
9a	Cierre de sesión	Página principal	Pagina principal	Correcto

Tabla 32: Pruebas relativas respecto al inicio y cierre de sesión



6.3. Pruebas relativas respecto a la gestión de las duchas

Caso	Características	Resultado esperado	Resultado Obtenido	Nota
10a	El usuario reserva una cita para la ducha	Obtener una cita para la ducha	Obtiene una cita para la ducha	Correcto
11a	El usuario reserva una segunda cita para la ducha	Mensaje de error	Obtiene una cita para la ducha	Error
11b	El usuario reserva una segunda cita para la ducha	Mensaje de error	Mensaje de error	Correcto
12a	El administrador ve las citas de las duchas	El administrador ve las citas de las duchas	Se muestran las citas	Correcto
13a	El usuario borra una cita para la ducha	Se elimina una cita para la ducha	No se elimina una cita para la ducha	Error
13b	El usuario borra una cita para la ducha	Se elimina una cita para la ducha	Se elimina una cita para la ducha	Correcto
14a	El administrador selecciona una o varias duchas como averiadas	Se marcan las duchas como averiadas	Se marcan las duchas como averiadas	Correcto

Tabla 33: Pruebas relativas respecto a la gestión de las duchas



6.4. Pruebas relativas respecto a la gestión de la lavandería

Caso	Características	Resultado esperado	Resultado Obtenido	Nota
15a	El usuario reserva una cita para la lavandería	Obtener una cita para la lavandería	Obtiene una cita para la lavandería	Correcto
15b	El usuario reserva una segunda cita para la lavandería	Mensaje de error	Obtiene una cita para la lavandería	Error
16a	El usuario reserva una segunda cita para la lavandería	Mensaje de error	Mensaje de error	Correcto
17a	El administrador ve las citas de la lavandería	El administrador ve las citas de la lavandería	Se muestran las citas	Correcto
18a	El usuario borra una cita para la lavandería	Se elimina una cita para la lavandería	Se elimina una cita para la lavandería	Correcto

Tabla 34: Pruebas relativas respecto a la gestión de la lavandería



6.5. Pruebas relativas respecto a la gestión de las clases

Caso	Características	Resultado esperado	Resultado Obtenido	Nota
16a	El usuario se apunta a una clase	Se apunte a una clase	El alumno se matricula directamente	Error
16b	El usuario se apunta a una clase	Se apunte a una clase	Se apunte a una clase	Correcto
17a	El administrador matricula a un alumno	Alumno matriculado	Alumno matriculado	Correcto
18a	El administrador desmatricula a un alumno	El administrador desmatricula a un alumno	El administrador desapunta a un alumno	Error
18b	El administrador desmatricula a un alumno	El administrador desmatricula a un alumno	El administrador desmatricula a un alumno	Correcto
19a	El administrador elimina una clase	El administrador elimina una clase	La clase se elimina	Correcto

Tabla 35: Pruebas relativas respecto a la gestión de las clases



7. Conclusiones y trabajo futuro

En este ultimo apartado se trataran las diferentes conclusiones obtenidas tras realizar el proyecto, se mostrara la verificación de los objetivos que se establecieron al comienzo del mismo y finalmente se hablara de las distintas mejoras que podría tener este proyecto en el futuro.

7.1. ¿Se ha cumplido los objetivos establecidos?

Al principio de este proyecto se establecieron distintos objetivos que se pretendían obtener tras la realización del proyecto, a continuación, se analizara si los siguientes objetivos se han cumplido:

- **Reducción de las colas:** Con la aplicación web implementada se le da la opción a los usuarios de poder reservar una cita para la ducha y la lavandería. Gracias a esta mejora, los usuarios podrán saber con anterioridad si hay huecos para las duchas o no, evitando hacer cola con tiempo de antelación. De esta forma, solo tendrán que acudir a la hora que tienen la cita para la ducha y si lo desean poder usar el servicio del café.
- **Reducir el impacto visual y acústico:** El cumplimiento de este objetivo, es la continuación del anterior. Una ve que las colas han sido reducidas y los usuarios ya disponen del servicio de reserva de citas para la ducha, el impacto visual y acústico se vera considerablemente reducido.
- **Incrementar el numero de usuarios que puedan usar el sistema:** Ya que se dispone de un sistema online de reservas, los usuarios que antes no podían antes de la apertura a hacer cola, ahora tienen la oportunidad de reservar una cita con anterioridad para poder hacer uso de los distintos servicios.
- **Reducir a dos días el servicio de lavandería:** Con el nuevo sistema de citas online para la lavandería se consigue reducir en un día todo el proceso de la lavandería. Ahora, el usuario puede reservar la cita vía online, concertando la cita para entregar la ropa y cuando esta este lista, los administradores le avisaran de que esta lista para que pase a recogerla.
- **Satisfacer a los miembros de Lagun Artean con el proyecto creado:** Gracias a esta mejora en su sistema y los objetivos anteriores cumplidos, los administradores gracias a esta mejora tendrán menos trabajo y podrán atender mejor a los usuarios del servicio.

7.2. El proyecto previsto y el proyecto finalizado

Como ya se ha comentado en el apartado de Problemas y soluciones del apartado de Desarrollo, en la implementación se ha demorado mas de lo esperado, en cambio, otras tareas han tardado menos, por lo que ha habido cambios en la planificación temporal inicial. Además, los cambios en la planificación final también modificaran la evaluación económica.



7.2.1. Planificación final

Tarea	Estimación inicial	Estimación final
1. Gestión	44 horas	40 horas
1.1. Reuniones Lagun Arteen	20 horas	-
1.2. Realizar el DOP	22 horas	20 horas
2. Análisis	115 horas	120 horas
2.1. Captura de requisitos	30 horas	-
2.2. Seleccionar herramientas	20 horas	-
2.3. Casos de Uso	35 horas	40 horas
2.4. Modelo de Dominio	10 horas	-
2.5. Transformación del Modelo a BBDD	20 horas	-
3. Diseño	25 horas	25 horas
3.1. Diagramas de Secuencia	25 horas	-
4. Implementación	180 horas	150 horas
4.1. Instalación de las herramientas	15 horas	8 horas
4.2. Instalación de la Base de Datos	15 horas	8 horas
4.3. Implementar funcionalidades comunes	50 horas	24 horas
4.4. Implementar funcionalidades administrador	50 horas	-
4.5. Implementar funcionalidades usuario	50 horas	60 horas
5. Plan de pruebas	60 horas	70 horas
5.1. Realizar plan de pruebas	30 horas	35 horas
5.2. Verificar pruebas	30 horas	35 horas
6. Documentación	120 horas	120 horas
6.1. Realizar memoria del TFG	115 horas	-
6.2. Realizar los manuales de uso	5 horas	-
Horas totales	544 horas	525 horas

Tabla 36: Planificación final de las tareas

Como podemos observar en el apartado de Gestión, la tarea de la realización del DOP ha sufrido un ligero cambio, reduciéndose su duración a 20 horas. En el apartado de Análisis, la tarea de Casos de Uso ha aumentado su duración de las 35 horas iniciales a las 40 horas.

Por otra parte, donde ha habido mas cambios ha sido en el apartado de Implementación donde las tareas de instalación se han reducido considerablemente y las implementación de las funcionalidades comunes es la que mas cambio ha tenido, reduciéndose mas de la mitad el tiempo estimado inicialmente.



Por ultimo, el apartado de plan de pruebas también ha sido modificado, aumentándose la estimación final de las tareas de 30 horas cada una a 35 horas.

7.2.2. Evaluación económica final

Debido a que ha sido modificada la duración del proyecto, la evaluación económica también ha sufrido cambios. Volviendo al razonamiento que hicimos en el apartado de evaluación económica: El salario se divide en 14 pagas, donde cada paga es de 1.712,42€/mes. Sabiendo que una jornada laboral en una semana son 40 horas y mensualmente son 160 horas, el precio por hora de un trabajador es de 10,70€/hora. Si a todo esto, lo multiplicamos por las 525 horas del proyecto, obtenemos que la mano de obra son 5.617,50€.

Concepto	Coste
Hardware	344,28€
Software	134,10€
Internet	783€
Luz eléctrica	7,28€
Mano de obra	5.617,50€
Total	6.886,16€

Tabla 37: Gastos totales del proyecto

7.3. Trabajo futuro

Al ser este un proyecto distinto en el que hay un cliente, es difícil definir las líneas de trabajo futuras de este proyecto ya que habría que tener en cuenta sus propuestas. Sin embargo, como ingenieros informáticos, siempre se puede proponer distintas mejoras para el proyecto actual.

Una posible mejora técnica, podría ser renovar la interfaz. Dado que la usan usuarios de distintos idiomas, podría ser algo más accesible si en vez de poner tantos idiomas, la interfaz funcionase con una mezcla de texto e iconos.

En la introducción del proyecto, hablamos sobre los distintos servicios que ofrecía Lagun Artean, entre ellos, el del centro de noche, el centro de día Kide o las viviendas tuteladas. Podría ser una buena oportunidad y aprovechando que la aplicación es fácilmente escalable, añadir los diferentes servicios que oferta la organización. Otra opción, sería ya que Lagun Artean ya disponía de una página web de información, unificar ambas plataformas en una única.



8. Referencias

- (1) BOE-A-2019-14977 sobre el Convenio colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos:
<https://www.boe.es/boe/dias/2019/10/18/pdfs/BOE-A-2019-14977.pdf>
- (2) Códigos QR:
<https://www.unitag.io/es/qrcode/what-is-a-qrcode>
- (3) API Google Charts:
<https://developers.google.com/chart>
- (4) API Webcam Capture:
<https://github.com/sarxos/webcam-capture>
- (5) BOE-A-2018-16673 sobre la Ley Orgánica de Protección de Datos personales y Garantía de los Derechos Digitales (LOPDGDD):
<https://www.boe.es/boe/dias/2018/12/06/pdfs/BOE-A-2018-16673.pdf>
- (6) MongoDB:
<https://www.genbeta.com/desarrollo/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>
- (7) Firebase:
<https://es.wikipedia.org/wiki/Firebase>
- (8) MySQL:
<https://es.wikipedia.org/wiki/MySQL>
- (9) HTTPS:
<https://es.ryte.com/wiki/HTTPS>
- (10) ¿Qué es Azure?:
<https://azure.microsoft.com/es-es/overview/what-is-azure/>
- (11) Digital Ocean:
<https://es.wikipedia.org/wiki/DigitalOcean>
- (12) Lagun Arteam:
<http://www.lagun-arteann.org/>
- (13) Ley Organica 4/2007:
<https://www.boe.es/eli/es/lo/2007/04/12/4/dof/spa/pdf>
- (14) Ley Orgánica 4/2007, de 12 de abril, por la que se modifica la Ley Orgánica 6/2001, de 21 de diciembre, de Universidades.:
<https://www.boe.es/eli/es/lo/2007/04/12/4/dof/spa/pdf>
- (15) Hacia una Educación Universitaria Transformadora, Proyectos Fin de Carrera en Cooperación al Desarrollo realizados en la UPV/EHU en el periodo 2006 - 2012:
<https://www.ehu.eus/documents/2957893/10539258/PFC-Cooperacion-Desarrollo.pdf/46bf6302-040f-2cf0-4200-cd28a990c549>



-
- (16) Código de conducta de las universidades en materia de Cooperación al Desarrollo:
<https://www.ehu.eus/documents/2957893/10539258/Codigo-Conducta.pdf/4797e31f-1936-fdc1-72c0-167042164b73>



Anexos

A. Anexo A: Casos de Uso extendidos

A.1. Registro

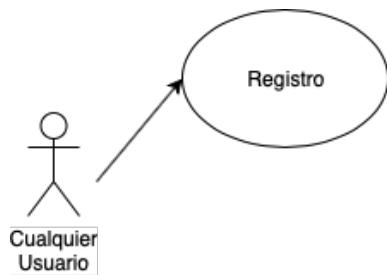


Figura 55: Caso de uso extendido: Registro

- **Nombre:** El usuario de Lagun Artean se registra en el sistema.
- **Descripción:** Permite a un usuario de las instalaciones de Lagun Artean, registrarse en el sistema para posteriormente acceder a las diferentes funcionalidades del sistema. Para que el registro sea completo, tendrá que ir a las instalaciones de Lagun Artean para completar el registro.
- **Actores:** Cualquier usuario de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se le muestra un menú con las opciones de inicio de sesión, registro, cambio de contraseña y ayuda. (Figura 56)
(Si el usuario pulsa en la opción de registro)
 - 1.1. El usuario introduce su nombre, apellidos y fecha de nacimiento.
(Figuras 57)
 - 1.2. Se le muestra un mensaje diciendo que debe acudir a las instalaciones para completar el registro. (Figura 58)
- **Postcondición:** Se almacenan parte de los datos del registro.



■ Interfaz gráfica:

- Aplicación web:

The screenshot shows a web browser window with the title "Lagun Artean". The URL bar contains "http://citas.lagun-artean.org". The main content area displays the "Lagun Artean" logo at the top, followed by four rectangular buttons: "Inicio Sesión", "Registro", "Cambio Contraseña", and "Ayuda".

Figura 56: Página principal Web

The screenshot shows a web browser window with the title "Lagun Artean". The URL bar contains "http://citas.lagun-artean.org/registro". The main content area displays a form for registration with fields for "Nombre" (labeled "Usuario"), "Apellidos" (labeled "Apellidos"), and "Edad" (labeled "Edad"). Below the fields is a note: "* Para completar el registro tendrás que venir a nuestras instalaciones". A "Aceptar" button is at the bottom right.

Figura 57: Registro Web



Figura 58: Completar registro Web



A.2. Inicio de sesión de usuarios

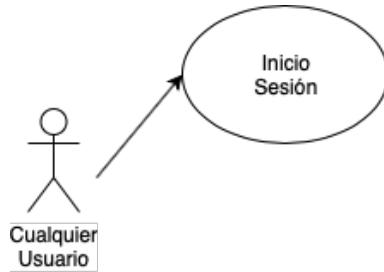


Figura 59: Caso de uso extendido: Inicio de sesión de usuarios

- **Nombre:** El usuario de Lagun Artean inicia sesión en el sistema.
- **Descripción:** Permite a un usuario de las instalaciones de Lagun Artean, tanto administrador como usuarios, acceder al sistema para posteriormente acceder a las diferentes funcionalidades del sistema.
- **Actores:** Cualquier usuario de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se le muestra un menú con las opciones de inicio de sesión, registro y ayuda. (Figura 56)
(Si el usuario pulsa en la opción de inicio de sesión)
 2. El usuario introduce su usuario y contraseña.(Figura 60)
(Si el usuario es incorrecto)
 - 2.1. Se le muestra un mensaje diciendo que el usuario o contraseña es incorrecto.
(Si la contraseña es incorrecta)
 - 2.2. Se le muestra un mensaje diciendo que la contraseña es incorrecta.
(Si el usuario no existe)
 - 2.3. Se le muestra un mensaje diciendo que debe registrarse para poder usar el servicio.
- **Postcondición:** Ninguna.



- **Interfaz gráfica:**

- **Aplicación web:**

A screenshot of a web browser showing a login form. The title bar says "Lagun Artean". The address bar shows the URL "http://citas.lagun-artean.org/iniciar_sesion". The main content area has a heading "Lagun Artean". It contains two input fields: one labeled "Usuario" and another labeled "Contraseña". Below these fields is a button labeled "Aceptar".

Figura 60: Inicio sesión Web



A.3. Cambio de contraseña

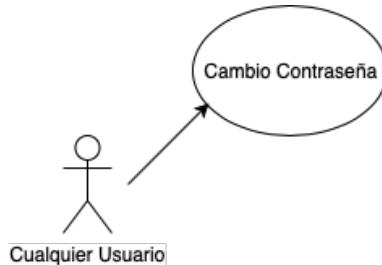


Figura 61: Caso de uso extendido: Ver usuarios

- **Nombre:** El usuario de Lagun Artean cambia la contraseña de acceso al sistema.
- **Descripción:** Permite al usuario de Lagun Artean cambiar la contraseña de acceso al sistema.
- **Actores:** Cualquier usuario de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se le muestra un menú con las opciones de inicio de sesión, registro, cambio de contraseña y ayuda. (Figura 56)
(Si el usuario pulsa en la opción de cambiar contraseña)
 - 1.1. El usuario selecciona la opción de cambiar contraseña.
 - 1.2. Introduce la nueva contraseña y la confirmación de la nueva contraseña y pulsa aceptar. (Figura 62)
 - 1.3. Se le muestra un mensaje por pantalla al usuario de que se ha cambiado correctamente la contraseña. (Figura 63)
- **Postcondición:** Se almacena la nueva contraseña.



■ Interfaz gráfica:

- Aplicación web:

The screenshot shows a web browser window for 'Lagun Artean'. The URL in the address bar is http://citas.lagun-artean.org/iniciar_sesion. The page title is 'Lagun Artean'. It contains two input fields labeled 'Nueva Contraseña' and 'Confirmar Contraseña', both with placeholder text. Below them is a 'Aceptar' button.

Figura 62: Cambio Contraseña Web

The screenshot shows a web browser window for 'Lagun Artean'. The URL in the address bar is http://citas.lagun-artean.org/cambio_contra. The page title is 'Lagun Artean'. A central message box displays the text 'La contraseña se ha cambiado correctamente.' Below it is an 'Aceptar' button.

Figura 63: Aviso cambio contraseña Web



A.4. Comprobar solicitudes de registro de usuarios

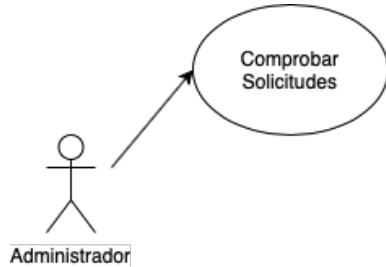


Figura 64: Caso de uso extendido: Comprobar solicitudes de registro de usuarios

- **Nombre:** El administrador de Lagun Artean ver las solicitudes de registro.
- **Descripción:** Permite al administrador de Lagun Artean comprobar las solicitudes de registro. Para que un usuario sea registrado y aceptado por el sistema, tiene que acceder a las instalaciones de Lagun Artean para completar su solicitud de registro.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** El administrador usa la versión web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. El administrador selecciona la opción de 'Comprobar Solicitudes'. (Figura 65)
 2. Se le muestra un listado con los usuarios pendientes de aprobar. (Figura 66)
 3. El administrador selecciona el usuario y pulsa 'Aceptar'.
 4. El administrador rellena los datos que hagan falta para que el registro se complete y pulsara 'Crear Usuario'. (Figura 67)
 5. Se genera un usuario y contraseña aleatorias para proporcionárselas al usuario. (Figura 68)
- **Postcondición:** Ninguna.



■ Interfaz gráfica:

- Aplicación web:

The screenshot shows a web browser window titled "Lagun Artean". The address bar contains the URL "http://citas.lagun-artean.org". The main content area is titled "Lagun Artean" and contains several buttons:

- Registro Usuario
- Comprobar Solicitudes
- Ver usuarios
- Ver citas ducha
- Ver citas lavanderia
- Ropa Lavanderia
- Gestionar Clases

Figura 65: Panel administrador Web

The screenshot shows a web browser window titled "Lagun Artean". The address bar contains the URL "http://citas.lagun-artean.org". The main content area is titled "Lagun Artean" and displays a table of registered users:

Nombre	Apellidos	Fecha nacimiento	Estado
Ada	Lovelace	December 10, 1815	Pendiente de aprobar
Grace	Hopper	December 9, 1906	Pendiente de aprobar
Margaret	Hamilton	August 17, 1936	Pendiente de aprobar
Joan	Clarke	June 24, 1917	Pendiente de aprobar

Aceptar

Figura 66: Registro final Web



Lagun Artean

http://citas.lagun-artean.org/registro

Lagun Artean

Nombre E-mail

Apellidos

Edad

Figura 67: Editar usuario Web

Lagun Artean

http://citas.lagun-artean.org/registro

Lagun Artean

Usuario Creado

Usuario U0001
Contraseña: xCT7uk

Figura 68: Usuario creado Web



A.5. Apuntarse a clases de castellano

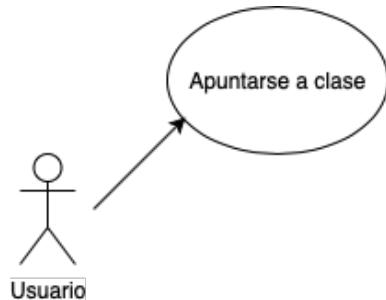


Figura 69: Caso de uso extendido: Apuntarse a clases de castellano

- **Nombre:** El usuario de Lagun Artean podrá apuntarse a clases de castellano.
- **Descripción:** Permite a un usuario de las instalaciones de Lagun Artean apuntarse a clases de castellano.
- **Actores:** Usuario de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. El usuario selecciona la opción de apuntarse a clase. (Figura 70)
 2. El usuario selecciona el grupo de clases que mas le convenga. (Figura 71)
(Si hay huecos libres)
 - 2.1. Se le muestra un mensaje de que tiene que pasarse por las instalaciones para confirmar el nivel al que se ha apuntado. (Figura 72)
(Si no hay huecos libres en la clase que ha seleccionado)
 - 2.2. Se le añade a una lista de espera del grupo que el usuario ha seleccionado. (Figura 73)
- **Postcondición:** Ninguna.



■ Interfaz gráfica:

- Aplicación web:

The screenshot shows a web page titled "Lagun Artean". At the top, there is a header bar with a back/forward button, a refresh icon, and the URL "http://citas.lagun-artean.org". Below the header, the page title "Lagun Artean" is displayed. A grid of six rectangular buttons is arranged in two rows of three. The buttons are labeled: "Cita Ducha" (top-left), "Ver citas ducha" (top-right), "Cita Lavandería" (middle-left), "Ver citas Lavandería" (middle-right), "Apuntarse a clase" (bottom-left), and "Cambio Contraseña" (bottom-right). All buttons have a light gray background and a thin black border.

Figura 70: Panel usuario Web

The screenshot shows a web page titled "Lagun Artean". At the top, there is a header bar with a back/forward button, a refresh icon, and the URL "http://citas.lagun-artean.org/clases". Below the header, the page title "Lagun Artean" is displayed. A table lists three class offerings. The first row contains: "Lunes - Miércoles 9:00 - 10:00" (radio button selected), "Alfabetización", and "1 hueco libre". The second row contains: "Martes - Jueves 9:00 - 10:00" (radio button unselected), "Medio", and "Grupo lleno". The third row contains: "Miércoles - Viernes 11:00 - 13:00" (radio button unselected), "Básico", and "Grupo lleno". At the bottom right of the table is a "Aceptar" (Accept) button.

Figura 71: Lista clases Web

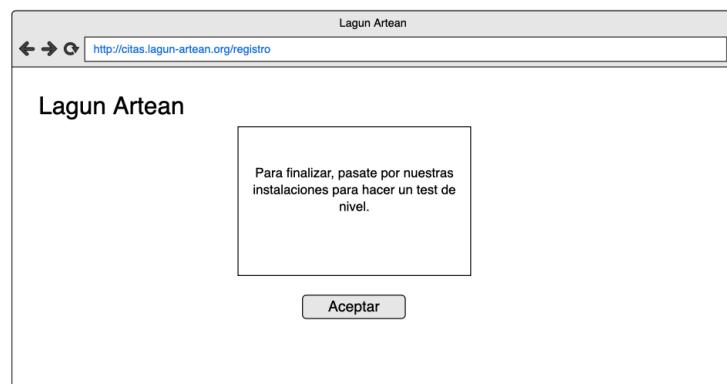


Figura 72: Aviso clases llenas Web

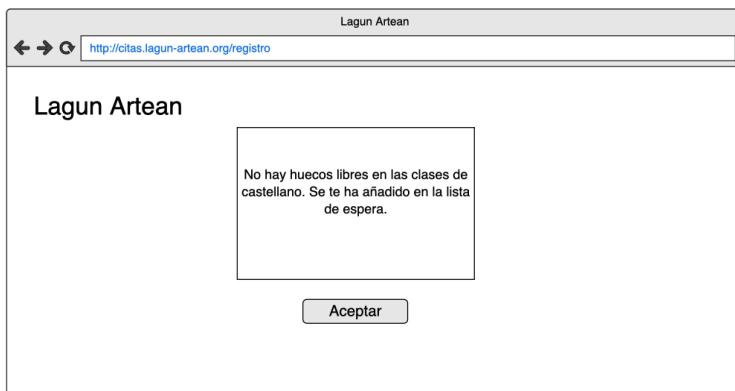


Figura 73: Aviso clases llenas Web



A.6. Ver ayuda

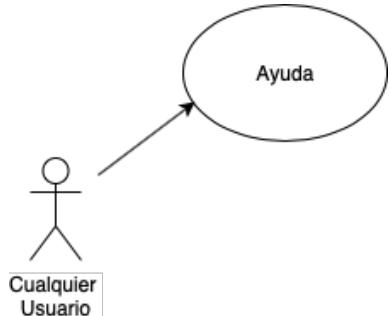


Figura 74: Caso de uso extendido: Ver ayuda

- **Nombre:** Ver ayuda del sistema.
- **Descripción:** Cualquier usuario puede solicitar ayuda para ver cómo funciona el sistema
- **Actores:** Cualquier usuario de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se le muestra un menú con las opciones de inicio de sesión, registro y ayuda. (Figura 56)
 2. El usuario pulsa en ayuda
 3. Se le muestra información sobre como funciona el sistema. (Figura 75)
- **Postcondición:** Ninguna.



- **Interfaz gráfica:**

- **Aplicación web:**

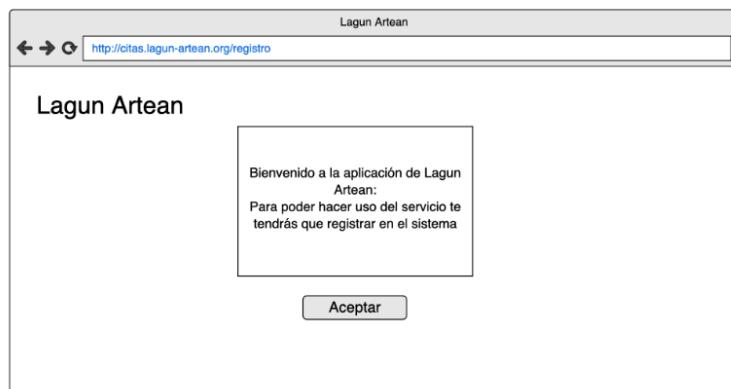


Figura 75: Ayuda Web



A.7. Pedir cita para las duchas

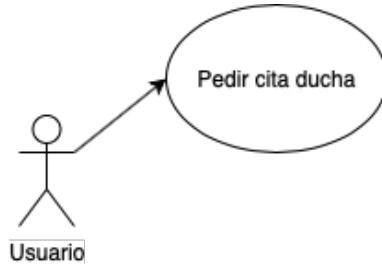


Figura 76: Caso de uso extendido: Pedir cita para las duchas

- **Nombre:** El usuario de Lagun Artean podrá pedir cita para las duchas.
- **Descripción:** Permite a un usuario de las instalaciones de Lagun Artean pedir cita para las duchas.
- **Actores:** Usuario de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran tres opciones: Cita Ducha, Cita Lavandería y Apuntarse a Clase. (Figura 70 y ??)
 2. El usuario le dará a 'Cita Ducha'.
 3. Se mostrará un calendario para que pueda elegir el día. (Figura 80 y ??)
(Si pulsa la X)
 - 3.1. Volverá al menú anterior.
 4. Saldrá una lista con el horario en el que puede reservar. (Figura 78 y ??)
 5. El usuario seleccionará una hora libre (las que están en verde).
(Si pulsa la X)
 - 5.1. No se guardará la cita y volverá al menú anterior.
 6. Le dará al tick y se guardará la cita. Saldrá un mensaje informando que se ha guardado.
- **Postcondición:** Se guarda la cita de la ducha.



■ Interfaz gráfica:

- Aplicación web:

The screenshot shows a web browser window for "Lagun Artean" at the URL <http://lagun-artean.org/citaDuchasCale>. The page title is "Lagun Artean". Below the title is a calendar for April 22, 2012. The calendar grid shows dates from 1 to 30. The date "22" is highlighted with a black border. At the bottom of the calendar are two buttons: one with a checkmark and one with a cross.

Figura 77: Calendario ducha Web

The screenshot shows a web browser window for "Lagun Artean" at the URL <http://lagun-artean.org/citaducha>. The page title is "Lagun Artean". Below the title is a booking schedule for April 22, 2019. The schedule lists time slots from 14:00 to 16:00 in 10-minute increments. Some slots are colored green, some red, and one slot at 14:50-15:00 has a checkmark. At the bottom of the schedule are two buttons: one with a checkmark and one with a cross.

Figura 78: Horario ducha Web



A.8. Pedir cita para la lavandería

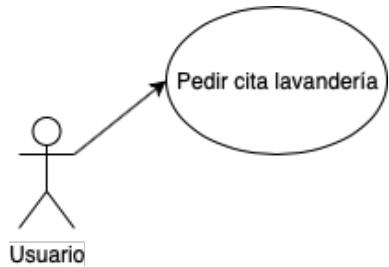


Figura 79: Caso de uso extendido: Pedir cita para la lavandería

- **Nombre:** El usuario de Lagun Arteen podrá pedir cita para la lavandería.
- **Descripción:** Permite a un usuario de las instalaciones de Lagun Arteen pedir cita para la lavandería.
- **Actores:** Usuario de Lagun Arteen.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran tres opciones: Cita Ducha, Cita Lavandería y Apuntarse a Clase. (Figura 70)
 2. El usuario le dará a Cita Lavandería.
 3. Se mostrará un calendario para que pueda elegir el día. (Figura 80)
(Si pulsa la (x))
 - 3.1. Volverá al menú anterior.
 4. El usuario seleccionará un día libre (estarán representados en el calendario). (Si pulsa la (x))
 - 4.1. No se guardará la cita y volverá al menú anterior.
 5. Le dará al tick y se guardará la cita. Saldrá un mensaje informando que se ha guardado.
- **Postcondición:** Se guarda la cita de la lavandería.



■ Interfaz gráfica:

• Aplicación web:

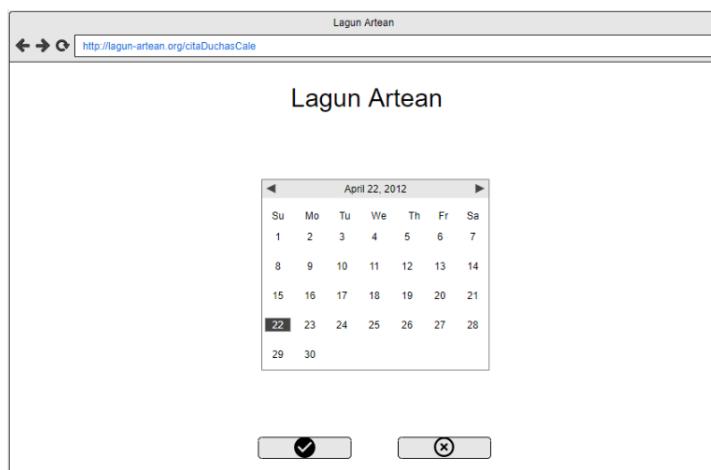


Figura 80: Calendario lavandería Web



A.9. Ver citas de la ducha

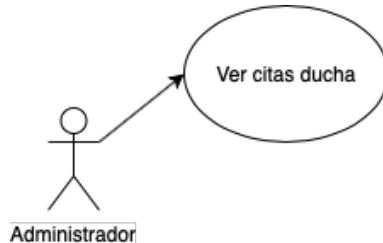


Figura 81: Caso de uso extendido: Ver citas de la ducha

- **Nombre:** El administrador de Lagun Artean podrá ver y editar las citas para la lavandería.
- **Descripción:** Permite a un administrador de las instalaciones de Lagun Artean ver y editar las citas para la lavandería.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicitudes, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas. (Figura 65)
 2. El administrador seleccionará Ver Citas Ducha.
 3. Saldrá el calendario para que elija día. (Figura 80)
(Si pulsa la (x))
 - 3.1. Volverá al menú anterior.
 4. Elegirá día y dará al tick.
 5. Se mostrarán las citas de ese día. (Figura 82) (Si da al lápiz)
 - 5.1. No se guardará la cita y volverá al menú anterior.
(Si da a la exportar a Excel)
 - 5.2. Tendrá la posibilidad de exportar las citas a Excel.
(Si da al tick)
 - 5.3. Se aplicarán los cambios y saldrá al menú.
(Si pulsa la (x))
 - 5.4. No se aplicarán los cambios y saldrá al menú.
- **Postcondición:** Ninguna.



■ Interfaz gráfica:

• Aplicación web:

A screenshot of a web browser window titled "Lagun Artean". The URL bar shows "http://lagun-artean.org/citaLavanderiaListaAdmin". The main content area has a header "Lagun Artean" and a date "22-04-2019". Below the date is a table with four rows of data. Each row contains a name in blue text and an edit icon (pencil and trash can) to its right. The names listed are Ada Lovelace, Margaret Hamilton, Juan Ortiz, and Anton Lopez. There are also several empty rows below them. At the bottom of the table are two buttons: one with a checkmark and one with a close symbol.

Figura 82: Lista ducha Web



A.10. Ver citas de la lavandería

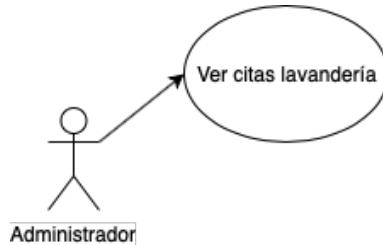


Figura 83: Caso de uso extendido: Ver citas de la lavandería

- **Nombre:** El administrador de Lagun Artean podrá ver y editar las citas para la lavandería.
- **Descripción:** Permite a un administrador de las instalaciones de Lagun Artean ver y editar las citas para las duchas.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicituds, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas.
 2. El administrador seleccionará Ver Citas Lavandería. (Figura 65)
 3. Saldrá el calendario para que elija día. (Figura 80)
(Si le da a la (x))
 - 3.1. Volverá al menú anterior.
 4. Elegirá día y dará al tick.
 5. Se mostrarán las citas de ese día. (Figura 84)
(Si da a exportar a Excel)
 - 5.1. Tendrá la posibilidad de exportar las citas a Excel.
(Si da al tick)
 - 5.1. Se aplicarán los cambios y saldrá al menú.
(Si da a la (x))
 - 5.1. No se aplicarán los cambios y saldrá al menú.
- **Postcondición:** Ninguna.



■ Interfaz gráfica:

- Aplicación web:

Ropa	Entregada
Ada Lovelace	Entregada
Margaret Hamilton	Entregada
Juan Ortiz	Entregada
Anton Lopez	Entregada

Añadir ✓ ✎

Figura 84: Lista Lavandería Web



A.11. Ver usuarios

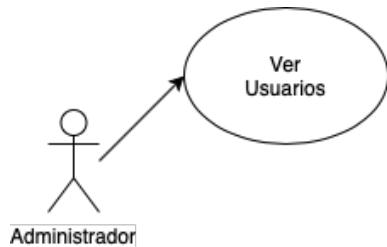


Figura 85: Caso de uso extendido: Ver usuarios

- **Nombre:** El administrador de Lagun Artean podra ver los usuarios de Lagun Artean y editarlos.
- **Descripción:** Permite al administrador de Lagun Artean ver los usuarios y si lo considera necesario, poder editarlos o eliminarlos.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** Ninguna.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicitudes, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas. (Figura 65)
 2. El administrador selecciona la opción de ver usuarios. (Figura 65)
 3. Se le muestra un listado de todos los usuarios del sistema. (Figura 86)
(Si elije un usuario y pulsa borrar)
 - 3.1. Se eliminara el usuario seleccionado
(Si elije un usuario y pulsa editar)
 - 3.1.1. El administrador podrá cambiar los campos que el administrador desee: usuario, contraseña, nombre...
(Si elije un usuario y pulsa aceptar)
 - 3.1.2. Vuelve al menú anterior
 - **Postcondición:** Se edita el usuario o se elimina el usuario si el administrador lo elige.
 - **Interfaz gráfica:**



• Aplicación web:

The screenshot shows a web application window titled "Lagun Artean". At the top, there is a header bar with a back/forward button, a refresh button, and a URL field containing "http://citas.lagun-artean.org". Below the header, the main title "Lagun Artean" is displayed. A table lists four users with columns for "Nombre", "Apellidos", and "Fecha nacimiento". The table has a scroll bar on the right side. At the bottom of the page are three buttons: "Borrar", "Editar", and "Aceptar".

Nombre	Apellidos	Fecha nacimiento
Ada	Lovelace	December 10, 1815
Grace	Hopper	December 9, 1906
Margaret	Hamilton	August 17, 1936
Joan	Clarke	June 24, 1917

Figura 86: Lista usuarios Web



A.12. Gestionar clases de castellano

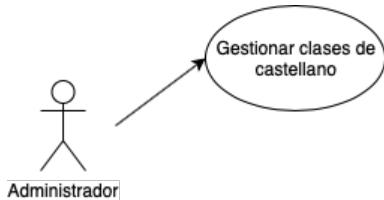


Figura 87: Caso de uso extendido: Gestionar clases de castellano

- **Nombre:** El administrador de Lagun Artean puede gestionar las clases de castellano.
- **Descripción:** El administrador de Lagun Artean tiene la opción de gestionar las clases de castellano, así como, añadir nuevas clases, matricular definitivamente a los usuarios una vez realizada la prueba de nivel.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** El administrador usa la versión web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicitudes, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas. (Figura 65)
 2. El administrador en el panel de administración selecciona la opción de Gestionar Clases. (Figura 65)
(Si el administrador desea gestionar los alumnos)
 - 2.1. Se abre una ventana con las opciones de Matricular y Desmatricular. (Figura 89)
(Si el administrador desea matricular alumnos)
 - 2.2. Se le abrirá una ventana con la lista de alumnos que quieren matricularse a uno de los grupos. (Figura 91)
 - 2.3. Si pulsa en un alumno y en Matricular el alumno se matriculará quedara matriculado dicho grupo.
(Si el administrador desea desmatricular alumnos)
 - 2.4. Se le abrirá una ventana con la lista de alumnos matriculados en cada grupo.
 - 2.5. Si pulsa en un alumno y en Desmatricular el alumno se desmatriculará de dicho grupo. (Figura 93)

(Si el administrador desea gestionar las clases)



-
3. Se abre una ventana con las opciones de Añadir Clase y Eliminar Clase. (Figura 90)
(Si el administrador desea añadir una clase)
 - 3.1. Se abre una ventana y añade la clase del día, hora y nivel que desee. (Figura 92)
(Si el administrador desea eliminar una clase)
 - 3.2. Se abre una ventana con las clases disponibles. (Figura 94)
 - 3.3. Selecciona una clase y pulsa en eliminar para que esa clase ya no exista, al eliminar la clase, también se eliminan los alumnos que estaba matriculados en ella.

■ **Postcondición:** Se guarda en el sistema la nueva clase o alumno apuntado.

■ **Interfaz gráfica:**

- **Aplicación web:**

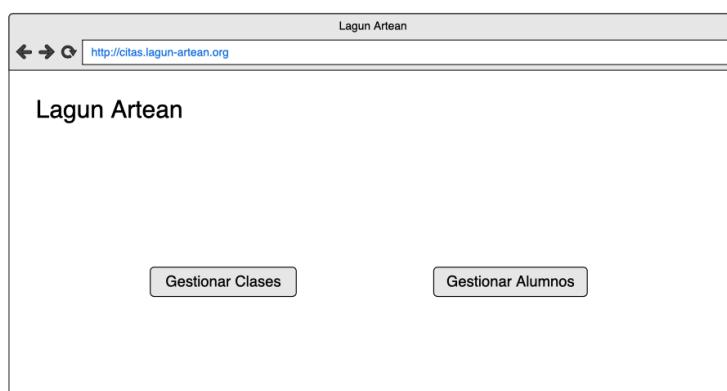


Figura 88: Menú gestión clases Web

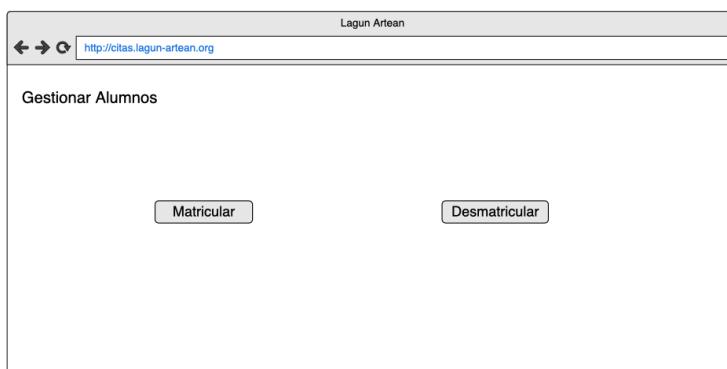


Figura 89: Menú matricula Web



Lagun Artean

<http://citas.lagun-artean.org>

Gestionar Clases

[Añadir Clase](#) [Eliminar Clase](#)

Figura 90: Menú clases Web

Lagun Artean

Solicitudes

Usuario	Nombre	Apellidos	Grupo	Nivel	Espera
U1001	Ada	Lovelace	Martes 10:00-11:00	Alfabetización	NO
U1231	Grace	Hopper	Lunes 10:00-11:00	Básico	SI
U1322	Maroare	Hamilton	Martes 11:00-12:00	Medio	NO
U1012	Joan	Clarke	Lunes 10:00-11:00	Básico	NO

Huecos

Grupo	Huecos
Martes 10:00-11:00	1
Lunes 10:00-11:00	1
Martes 11:00-12:00	2

[Matricular](#)

Figura 91: Matricular Web

Lagun Artean

<http://citas.lagun-artean.org>

Añadir Clase

Días	Horario	Nivel	Huecos
<input checked="" type="radio"/> Lunes	<input checked="" type="radio"/> 9:00 - 10:00	<input checked="" type="radio"/> Alfabetización	<input type="text" value="5"/> <input type="button" value="▼"/>
<input type="radio"/> Martes	<input type="radio"/> 10:00 - 11:00	<input type="radio"/> Básico	
<input type="radio"/> Miércoles	<input type="radio"/> 11:00 - 12:00	<input type="radio"/> Medio	
<input type="radio"/> Jueves	<input type="radio"/> 12:00 - 13:00		
<input type="radio"/> Viernes			

[Añadir](#)

Figura 92: Añadir clase Web



Lagun Artean

http://citas.lagun-artean.org

Alumnos Matriculados

Nombre	Apellidos	Grupo	Nivel
Ada	Lovelace	Lunes 10:00-11:00	Alfabetización
Grace	Hopper	Miércoles 10:00-11:00	Medio
Margaret	Hamilton	Lunes 10:00-11:00	Alfabetización
Joan	Clarke	Martes 10:00-11:00	Medio

Eliminar

Figura 93: Desmatricular alumno Web

Lagun Artean

http://citas.lagun-artean.org

Clases

Grupo	Nivel
Lunes 10:00-11:00	Alfabetización
Miércoles 10:00-11:00	Medio
Lunes 10:00-11:00	Alfabetización
Martes 10:00-11:00	Medio

Eliminar

Figura 94: Eliminar clase Web



A.13. Ver ropa

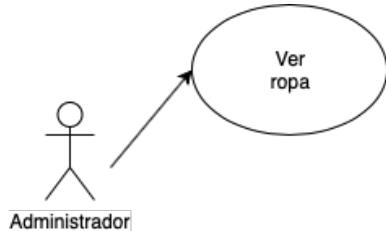


Figura 95: Caso de uso extendido: Ver ropa

- **Nombre:** El administrador de Lagun Artean podrá ver y editar la ropa que tienen de la lavandería.
- **Descripción:** Permite a un administrador de las instalaciones de Lagun Artean ver y editar las citas para la lavandería.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** El administrador usa la versión web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicitudes, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas. (Figura 65)
 2. El administrador selecciona Ropa Lavandería.
 3. Se mostrará la ropa que tienen en el establecimiento. (Figura 96)
(Si da a entregada)
 - 3.1. La ropa se eliminara de la pantalla. (Si da a añadir)
 - 3.2. Podrá añadir a mano ropa que hayan recogido. (Si da al tick)
 - 3.3. Podrá añadir a mano ropa que hayan recogido. (Si da a la (x))
 - 3.4. Se aplicarán los cambios y saldrá al menú.
- **Postcondición:** Ninguna.



■ Interfaz gráfica:

- Aplicación web:

The screenshot shows a Mozilla Firefox browser window with the URL <http://moqups.com> in the address bar. The main content area displays a web application titled "Lagun Artean". A table lists four items under the heading "Ropa": Ada Lovelace, Margaret Hamilton, Juan Ortiz, and Anton Lopez. To the right of each item is a status button labeled "Entregada". Below the table is a "Añadir" button. At the bottom are two buttons: one with a checkmark and another with an X.

Ropa	
Ada Lovelace	Entregada
Margaret Hamilton	Entregada
Juan Ortiz	Entregada
Anton Lopez	Entregada

Añadir

✓ ✘

Figura 96: Lista Ropa Web



A.14. Ver estadísticas sobre el sistema

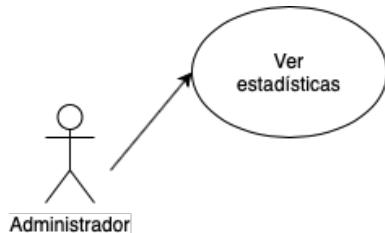


Figura 97: Caso de uso extendido: Ver estadísticas sobre el sistema

- **Nombre:** El administrador de Lagun Artean podrá ver las estadísticas sobre el uso del servicio.
- **Descripción:** Permite a un administrador de las instalaciones de Lagun Artean ver las estadísticas del servicio.
- **Actores:** Administrador de Lagun Artean.
- **Precondición:** El administrador usa la versión web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicitudes, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas. (Figura 65)
 2. El administrador selecciona Estadísticas.
 3. Se mostrará un listado sobre las diferentes estadísticas a mostrar. (Figura 98)
 - 3.1. (Si da a uso de las duchas por sexo)
Se mostrarán el uso de las duchas por sexo.
 - 3.2. (Si da a uso de las duchas por nacionalidad)
Se mostrarán el uso de las duchas por nacionalidad.
 - 3.3. (Si da a uso total de las duchas por año)
Se mostrarán el uso de las duchas por año.
 - 3.4. (Si da a uso total de la lavandería por año)
Se mostrarán el uso de la lavandería por año.
 - 3.5. (Si da a uso de la lavandería por nacionalidad)
Se mostrarán el uso de la lavandería por nacionalidad.



-
- 3.6. (Si da a uso de la lavandería por nacionalidad y sexo)
Se mostrarán el uso de la lavandería por nacionalidad y sexo.
- 3.7. (Si da a numero de usuarios con documentación)
Se mostrarán el numero de usuarios que poseen documentación.
- 3.8. (Si da a numero de usuarios con documentación y por nacionalidad)
Se mostrarán el numero de usuarios que poseen documentación.

■ **Postcondición:** Ninguna.

■ **Interfaz gráfica:**

- **Aplicación web:**

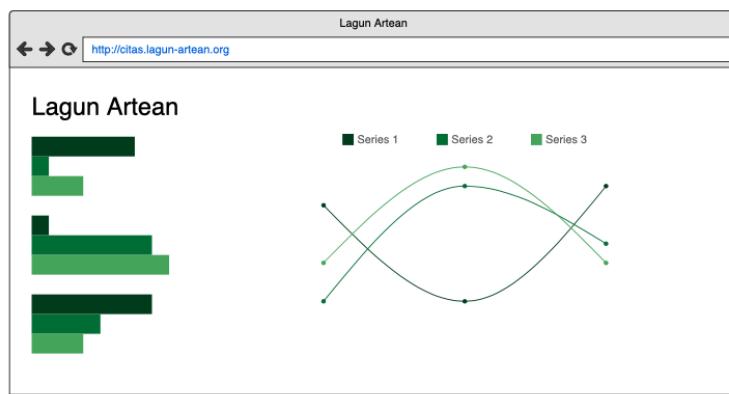


Figura 98: Estadísticas



A.15. Apuntarse al café

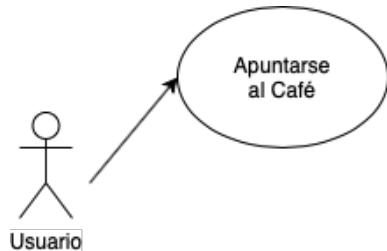


Figura 99: Caso de uso extendido: Apuntarse al café

- **Nombre:** El usuario de Lagun Artean se apunta a la cola para tomar el café.
- **Descripción:** Permite a un usuario de las instalaciones de Lagun Artean poder apuntarse a la cola del café.
- **Actores:** Usuario de Lagun Artean.
- **Precondición:** El usuario usa la aplicación de escritorio para poder leer códigos QR.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. El usuario acerca su teléfono móvil o un papel con el código QR que le identifica a la webcam del ordenador (Figura: 100).
 2. El usuario pulsa en el botón 'Escanear Código QR'
 3. El usuario recoge el ticket con su turno para el café para ese día en concreto.
- **Postcondición:** Ninguna.



- **Interfaz gráfica:**

- **Aplicación de escritorio:**

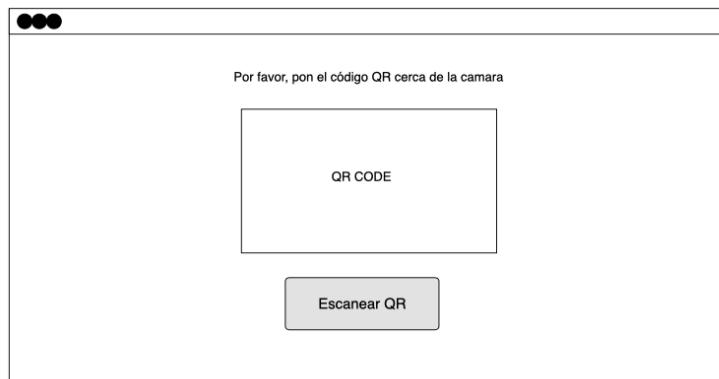


Figura 100: Escaneo Código QR



A.16. Mantenimiento duchas

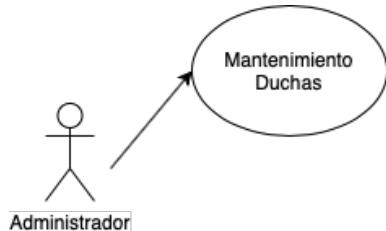


Figura 101: Caso de uso extendido: Mantenimiento duchas

- **Nombre:** El administrador de Lagun Artea selecciona las duchas que están averiadas o se han arreglado.
- **Descripción:** Permite al administrador de las instalaciones de Lagun Artea poder seleccionar las duchas que se han averiado o las duchas que se han arreglado.
- **Actores:** Administrador de Lagun Artea.
- **Precondición:** El administrador usa la aplicación web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran nueve opciones: Registro Usuario, Comprobar Solicituds, Ver Usuarios, Ver Citas Ducha, Ver Citas Lavandería, Ropa Lavandería, Mantenimiento Duchas, Gestionar Clases y Estadísticas. (Figura 65)
 2. El administrador selecciona 'Mantenimiento Duchas'.
 3. Se mostrara un listado con las 3 duchas del sistema (Figura: 102)
 - 3.1. (Si selecciona la ducha 1)
Se marcará la ducha 1 como averiada si se encontraba en funcionamiento o se marcará como en funcionamiento si se encontraba averiada.
 - 3.2. (Si selecciona la ducha 2)
Se marcará la ducha 2 como averiada si se encontraba en funcionamiento o se marcará como en funcionamiento si se encontraba averiada.
 - 3.3. (Si selecciona la ducha 3)
Se marcará la ducha 3 como averiada si se encontraba en funcionamiento o se marcará como en funcionamiento si se encontraba averiada.



■ **Postcondición:** Ninguna.

■ **Interfaz gráfica:**

- **Aplicación web:**

The screenshot shows a web browser window titled "Lagun Artean" with the URL "http://citas.lagun-artean.org". The page displays a table with three rows, each representing a shower (Ducha 1, Ducha 2, Ducha 3) and its current state ("Estado"). A vertical scroll bar is visible on the right side of the table. Below the table, there are three radio buttons labeled "Ducha 1", "Ducha 2", and "Ducha 3", each associated with a button labeled "Averiada" or "Funcionamiento".

Ducha	Estado
1	En Funcionamiento
2	En Funcionamiento
3	En Funcionamiento

Ducha 1

Ducha 2

Ducha 3

Figura 102: Mantenimiento Duchas



A.17. Ver reservas ducha

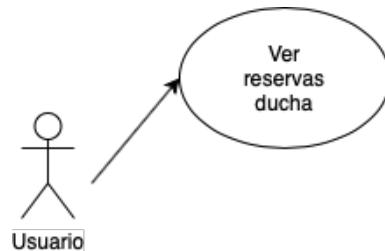


Figura 103: Caso de uso extendido: Mantenimiento duchas

- **Nombre:** El usuario de Lagun Artean ve las citas de las duchas.
- **Descripción:** Permite al usuario ver las citas de la ducha que tiene reservadas.
- **Actores:** Usuario de Lagun Artean.
- **Precondición:** El usuario usa la aplicación web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran cinco opciones: cita ducha, cita lavandería, clases de castellano, ver cita ducha, ver cita lavandería. (Figura 70)
 2. El usuario selecciona 'Ver citas ducha'.
 3. Se mostrara un listado con las citas para la ducha (Figura: 82)
(Si selecciona la fecha de la cita y le da a borrar)
- 3.1. Se borrara la cita
- **Postcondición:** Ninguna.



A.18. Ver reservas lavandería

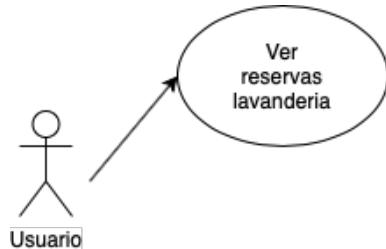


Figura 104: Caso de uso extendido: Ver reservas lavandería

- **Nombre:** El usuario de Lagun Artean ve las reservas que tiene para usar la lavandería.
- **Descripción:** Permite al usuario ver las citas de la lavandería que tiene reservadas.
- **Actores:** Usuario de Lagun Artean.
- **Precondición:** El usuario usa la aplicación web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran cinco opciones: cita Ducha, cita lavandería, clases de castellano, ver cita ducha, ver cita lavandería. (Figura 70)
 2. El usuario selecciona 'Ver citas lavandería'.
 3. Se mostrara un listado con las citas para la lavandería (Figura: 84)
(Si selecciona la fecha de la cita y le da a borrar)
 - 3.1. Se borrara la cita
- **Postcondición:** Ninguna.



A.19. Ver notificaciones

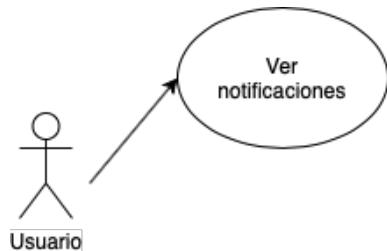


Figura 105: Caso de uso extendido: Ver notificaciones

- **Nombre:** El usuario de Lagun Artean ve las notificaciones
- **Descripción:** Permite al usuario ver las notificaciones, en las notificaciones pueden aparecer diferentes mensajes como: Hoy tienes una ducha reservada, hoy tienes una cita en la lavandería, tienes ropa por recoger en la lavandería.
- **Actores:** Usuario de Lagun Artean.
- **Precondición:** El usuario usa la aplicación web.
- **Requisitos no funcionales:** Ninguno.
- **Flujo de eventos:**
 1. Se muestran cinco opciones: cita Ducha, cita lavandería, clases de castellano, ver cita ducha, ver cita lavandería y notificaciones. (Figura 70)
 2. El usuario selecciona 'Notificaciones'.
 3. Se mostrarán las notificaciones del sistema (Figura: 70)
- **Postcondición:** Ninguna.



B. Anexo B : Diagramas de Secuencia

B.1. Diagrama de Secuencia: Registro

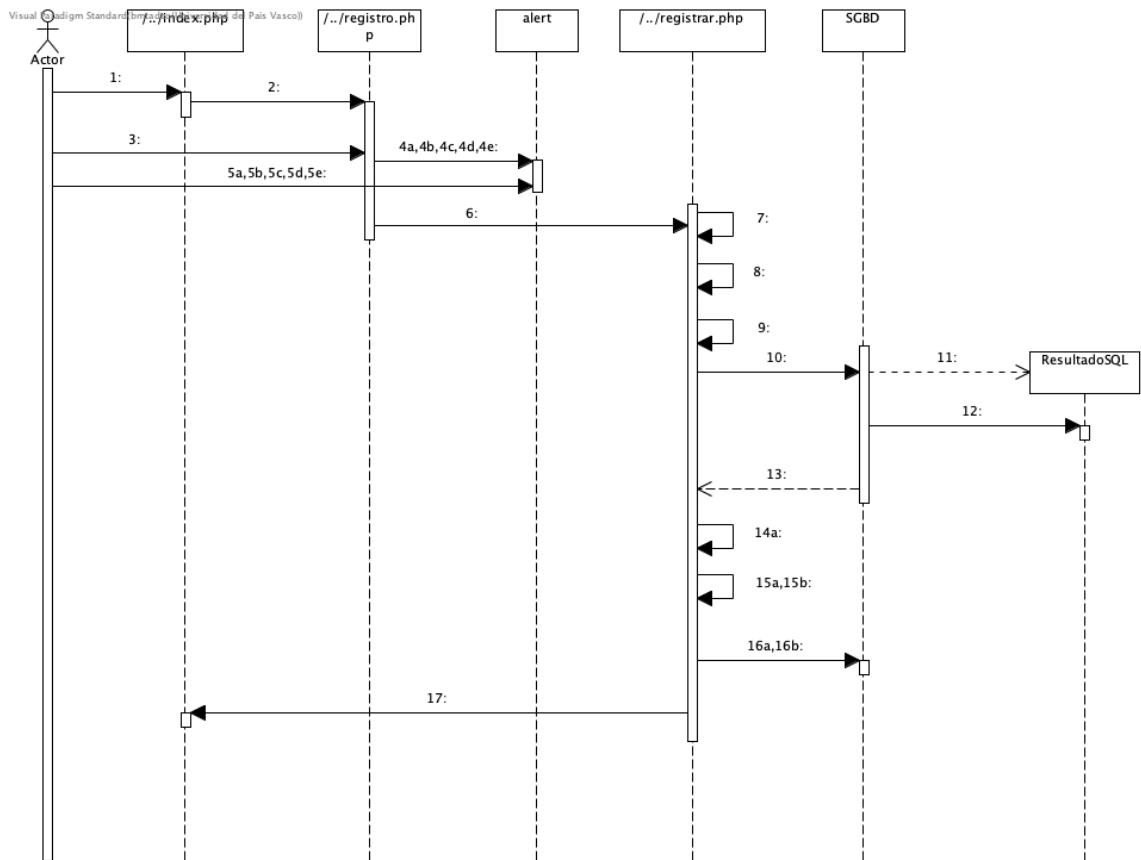


Figura 106: Diagrama de Secuencia: Registro

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Registro'.
- 2.- new ../registro.php
- 3.- El usuario escribe su nombre, apellido 1, su apellido 2, su documentación si tiene, su sexo, su nacionalidad, fecha de nacimiento y pulsa en 'Registrar'.
- (Si el nombre esta vacío)
 - 4a.- new alert()
 - 5a.- El usuario pulsa 'Aceptar'
- (Si el apellido 1 esta vacío)
 - 4b.- new alert()
 - 5b.- El usuario pulsa 'Aceptar'
- (Si el apellido 2 esta vacío)
 - 4c.- new alert()
 - 5c.- El usuario pulsa 'Aceptar'



(Si la fecha de nacimiento esta vacía)

 4d.- new alert()

 5d.- El usuario pulsa 'Aceptar'

(Si el sexo esta vacío)

 4e.- new alert()

 5e.- El usuario pulsa 'Aceptar'

6.- Método POST ->parámetros: \$nombre, \$apellido1, \$apellido2, \$nacimiento, \$sexo, \$documentación y \$nacionalidad ->registrar.php

7.- ramdomUser(): \$user

8.- ramdomPass(): \$pass

9.- existeUsuario(\$conexion, \$nombre, \$apellido1, \$apellido2, \$nacimiento, \$sexo, \$documentación ,\$user, \$pass, \$nacionalidad,\$qr)

10.- execSQL(SELECT * FROM Usuario WHERE nick='\$_user')

11.- new

12.- \$query=\$this->db->query(\$sql)->result()

13.- return \$query->result()

(Si row["nick"]==\$_user)

 14a.-ramdomUser(): \$user

 15a.- registrar(\$conexion, \$nombre, \$apellido1, \$apellido2, \$nacimiento, \$sexo, \$documentación, \$user, \$pass, \$nacionalidad,\$qr)

 16a.- execSQL(INSERT INTO Usuario(nacimiento, documentacion,nick, contrasenya, nombre, apellido1, apellido2, nacionalidad, sexo, qr) VALUES (\$nacimiento, \$documentacion, \$user, \$pass, \$nombre, \$apellido1, \$apellido2,\$nacionalidad,\$sexo,\$rcode))

 17a.- header("Location: ../../index.php");

(Si no)

 15b.- registrar(\$nombre, \$apellido1, \$apellido2, \$nacimiento, \$sexo, \$documentación ,\$_user, \$pass, \$nacionalidad,\$qr)

 16b.- execSQL(INSERT INTO Usuario(nacimiento, documentacion,nick, contrasenya, nombre, apellido1, apellido2, nacionalidad, sexo, qr) VALUES (\$nacimiento, \$documentacion, \$_user, \$pass, \$nombre, \$apellido1, \$apellido2,\$nacionalidad,\$sexo,\$rcode))

 17b.- header("Location: ../../index.php");



B.2. Diagrama de Secuencia: Inicio de Sesión

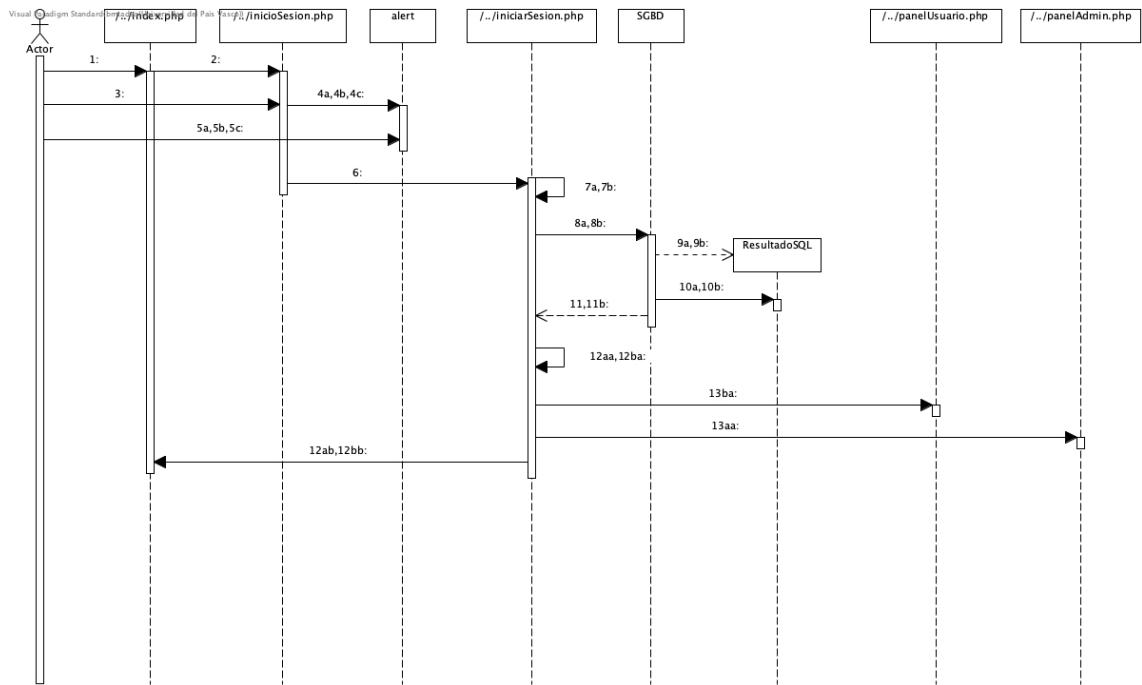


Figura 107: Diagrama de Secuencia: Inicio Sesión

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Iniciar Sesión'.
- 2.- new ../inicioSesion.php
- 3.- El usuario inserta su usuario y contraseña.
(Si el usuario o la contraseña están vacíos)
 - 4a.- new alert()
 - 5a.- El usuario pulsa 'Aceptar'
- (Si el usuario supera los 50 caracteres)
 - 4b.- new alert()
 - 5b.- El usuario pulsa 'Aceptar'
- (Si la contraseña supera los 50 caracteres)
 - 4c.- new alert()
 - 5c.- El usuario pulsa 'Aceptar'
- 6.- Método POST parámetros ->\$usuario, \$contraseña ->iniciarSesion.php
(Si el usuario es 'admin')
 - 7a.- loginAdmin(\$conexion, \$usuario, \$contraseña)
 - 8a.- execSQL(SELECT * FROM Administrador WHERE nick='\$usuario' and contraseña = '\$contraseña')
- 9a.- new
- 10a.- \$query=\$this->db->query(\$sql)->result()
- 11a.-return \$query->result()



```
(Si $num_rows ==1)
    12aa.- $_SESSION['usu_id']=$usuario
        13aa.- header("Location: ../../panelAdmin.php");
(Si $num_rows !=1)
    12ab.- header("Location: ../../index.php");
(Si no)
    7b.- loginAdmin($conexion, $usuario, $contrasenya)
        8b.- execSQL(SELECT * FROM Usuario WHERE nick='$usuario' and
contrasenya = '$contrasenya')
        9b.- new
        10b.-$query=$this->db->query($sql)->result()
        11b.-return $query->result()
(Si $num_rows ==1)
    12ba.- $_SESSION['usu_id']=$usuario
        13ba.- header("Location: ../../panelUsuario.php");
(Si $num_rows !=1)
    12bb.- header("Location: ../../index.php");
```



B.3. Diagrama de Secuencia: Cambio de Contraseña

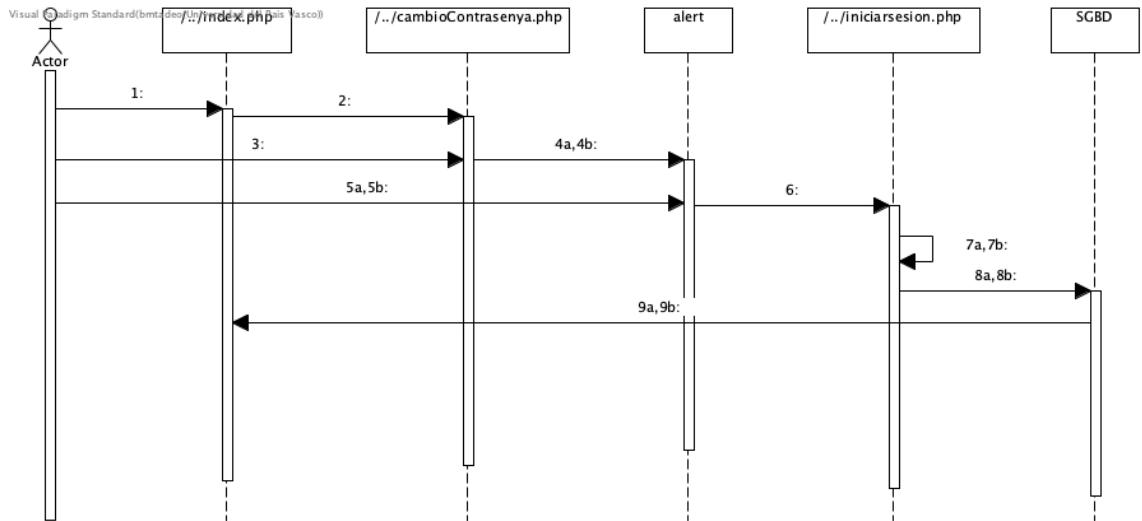


Figura 108: Diagrama de Secuencia: Cambio de Contraseña

■ Flujo de eventos:

- 1.- El usuario selecciona 'Cambiar Contraseña'.
- 2.- new ../cambioContrasena.php
- 3.- El usuario introduce su usuario, su antigua contraseña y su nueva contraseña.
(Si no introduce alguno de los campos requeridos)
 - 4a.- new alert()
 - 5a.- El usuario pulsa 'Aceptar'
- 4b.- new alert()
(Si la contraseña es de menos de 8 caracteres)
 - 5b.- El usuario pulsa 'Aceptar'
- 6.- Método POST parámetros ->\$usuario, \$contraseniaAnt, \$contraseniaNue ->iniciarSesion.php
(Si \$usuario =='admin')
 - 7a.- actualizarContraAdmin(\$conexion, \$usuario, \$contraseniaAnt, \$contraseniaNue)
 - 8a.- execSQL(UPDATE Administrador SET contrasenya='\$contraseniaNue' WHERE nick='\$usuario' AND contrasenya='\$contraseniaAnt')
 - 9a.- header("Location: ../index.php")
 - 7b.- actualizarContraUsuario(\$conexion, \$usuario, \$contraseniaAnt, \$contraseniaNue)
 - 8b.- execSQL(UPDATE Usuario SET contrasenya='\$contraseniaNue' WHERE nick='\$usuario' AND contrasenya='\$contraseniaAnt')
 - 9b.- header("Location: ../index.php")



B.4. Diagrama de Secuencia: Ver ayuda

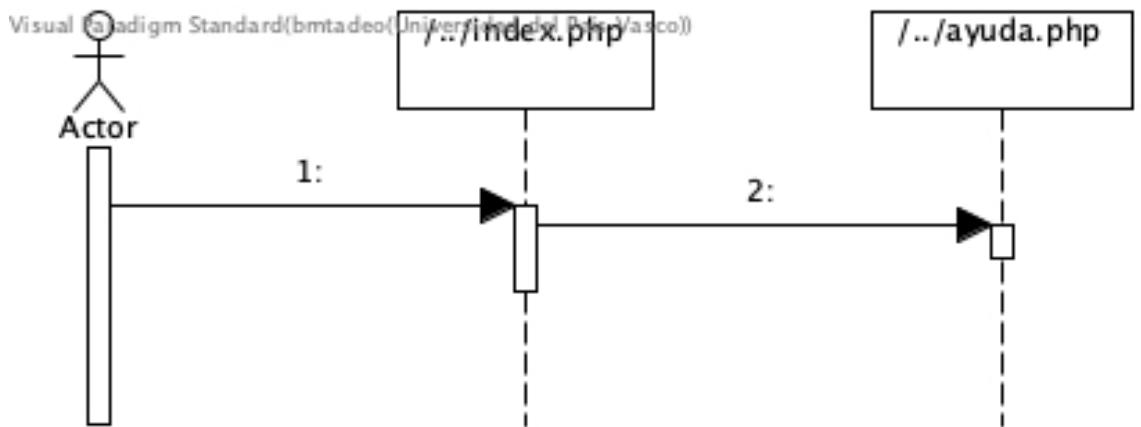


Figura 109: Diagrama de Secuencia: Ayuda

■ **Flujo de eventos:**

- 1.- El usuario pulsa en ayuda
- 2.- new ../ayuda.php



B.5. Diagrama de Secuencia: Comprobar solicitudes de registro de usuarios

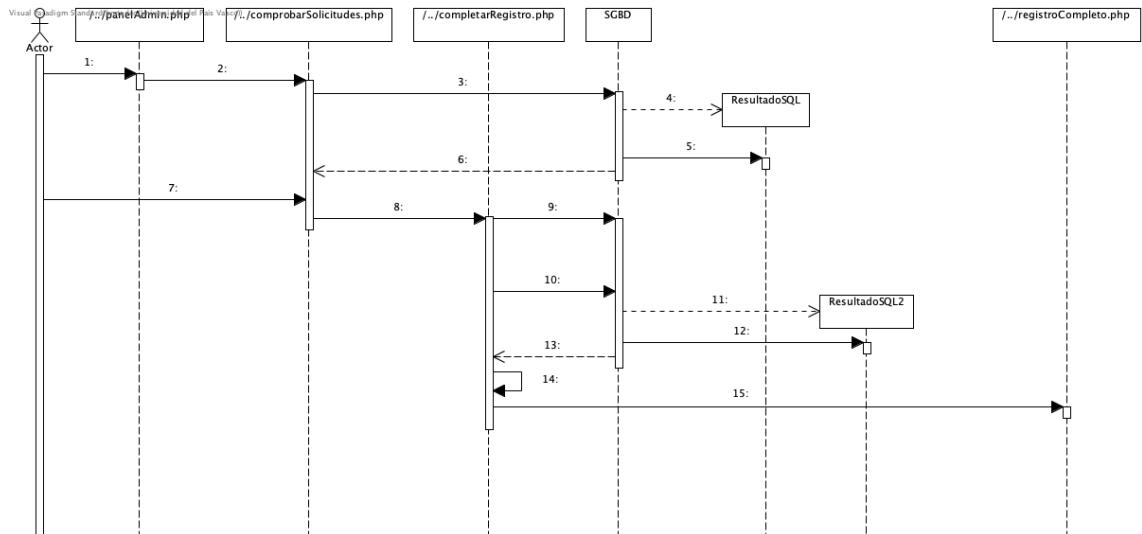


Figura 110: Diagrama de Secuencia: Comprobar solicitudes de registro de usuarios

■ Flujo de eventos:

- 1.- El administrador pulsa en 'Comprobar Solicitudes'
- 2.- new ../comprobarSolicitudes.php
- 3.- execSQL(SELECT * FROM Usuario where registroCompleto is false;) 4.- new 5.- \$query=\$this->db->query(\$sql)->result()
- 6.- return \$query->result()
- 7.- El usuario selecciona el usuario para completar el registro y rellena los datos necesarios para completarlo
- 8.- Método POST parámetros->\$usuario, \$contrasenya, \$telefono, \$email ->completarRegistro.php
- 9.- execSQL(UPDATE Usuario SET telefono ='\$telef',email ='\$email',registroCompleto =true WHERE nick= '\$nick');
- 10.- execSQL(SELECT qr FROM Usuario WHERE nick ='\$nick';)
- 11.- new
- 12.- \$query=\$this->db->query(\$sql)->result()
- 13.- return \$query->result()
- 14.- \$_SESSION["qr"]=\$row["qr"]
- 15.- header("Location: ../registroCompleto.php");



B.6. Diagrama de Secuencia: Apuntarse a clase de castellano

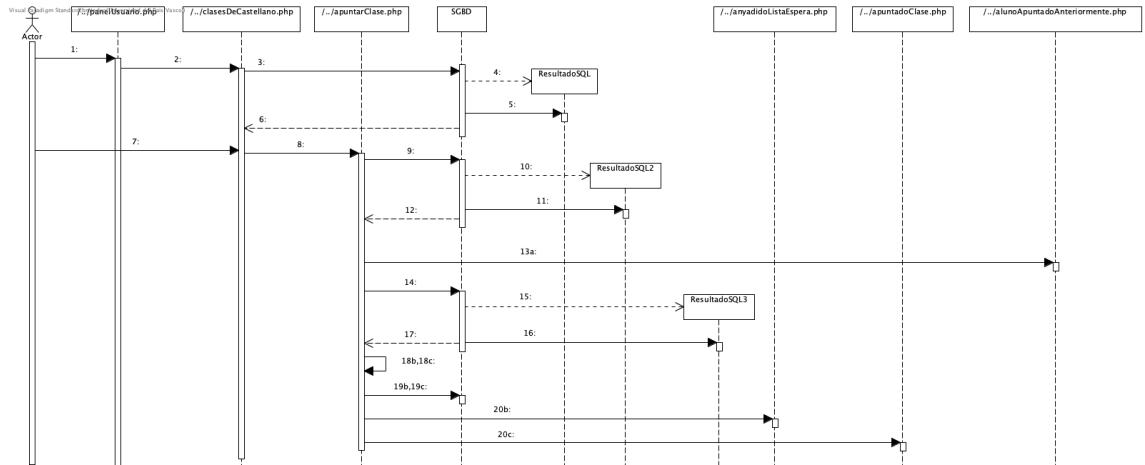


Figura 111: Diagrama de Secuencia: Apuntarse a clase de castellano

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Clases de Castellano'.
- 2.- new ../clasesDeCastellano.php
- 3.- execSQL(SELECT * FROM Clases);
- 4.- new
- 5.- \$query=\$this->db->query(\$sql)->result()
- 6.- return \$query->result()
- 7.- El usuario elige la clase de castellano que desea apuntarse.
- 8.- Método POST parámetros->\$grupo ->apuntarClase.php
- 9.- execSQL(SELECT idUsuario FROM Apuntado WHERE Apuntado.idClase = '\$grupo' AND idUsuario = '\$usuario');
- 10.- new
- 11.- \$query=\$this->db->query(\$sql)->result()
- 12.- return \$query->result()
- (Si \$row['idUsuario']==\$usuario)
 - 13a.- header("Location: ../../alumnoApuntadoAnteriormente.php");
 - 14.- execSQL(SELECT capacidad FROM Clases WHERE idClase = '\$grupo');
 - 15.- new
 - 16.- \$query=\$this->db->query(\$sql)->result()
 - 17.- return \$query->result()
 - (Si \$row['capacidad'] menor o igual a 0)
 - 18b.- insertApuntados(\$listaEspera, \$usuario, \$idGrupo, \$conexion);
 - 19b.- execSQL(INSERT INTO Apuntado (idUsuario, idClase, listaEspe-
ra) VALUES ('\$usuario', '\$idGrupo', true))
 - 20b.- header("Location: ../../anyadidoListaEspera.php");
 - (Si \$row['capacidad'] mayor a 0)
 - 18c.- insertApuntados(\$listaEspera, \$usuario, \$idGrupo, \$conexion);



```
19c.- execSQL(INSERT INTO Apuntado (idUsuario, idClase, listaEspe-
ra) VALUES ('$usuario','$idGrupo',false))
20c.- header("Location: ../../apuntadoClase.php");
```



B.7. Diagrama de Secuencia: Pedir cita para las duchas

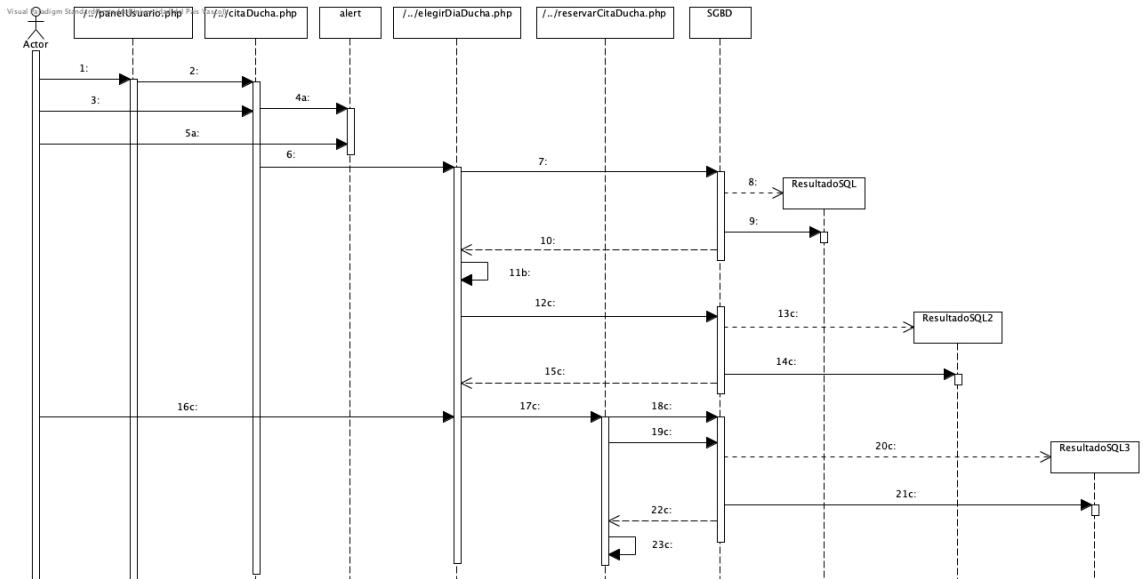


Figura 112: Diagrama de Secuencia: Pedir cita para las duchas

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Cita Ducha'.
- 2.- new ../citaDucha.php
- 3.- El usuario selecciona fecha.
(Si la fecha seleccionada es o sábado o domingo)
 - 4a.- new alert('Lo sentimos, nuestras instalaciones no abren los sábados y domingos')
 - 5a.- El usuario pulsa 'Aceptar'
- 6.- Método POST parámetros->\$fecha ->elegirDiaDucha.php
- 7.- execSQL(SELECT Usuario.idUsuario FROM Usuario WHERE '\$usu_id' IN (SELECT idUsuario FROM Duchan WHERE Duchan.fecha ='\$fecha'))
- 8.- new
- 9.- \$query=\$this->db->query(\$sql)->result()
- 10.- return \$query->result()
(Si \$usu_id!='admin' y \$usu_id==\$row['idUsuario'])
- 11b.- echo 'Ya has elegido ducha para este día.'
- (Si no)
- 12c.- execSQL(SELECT Ducha.idDucha, Ducha.hora FROM Ducha WHERE Ducha.averiada IS FALSE AND Ducha.idDucha NOT IN (SELECT Duchan.idDucha FROM Duchan WHERE Duchan.fecha ='\$fecha'))
- 13c.- new
- 14c.- \$query=\$this->db->query(\$sql)->result()
- 15c.- return \$query->result()
- 16c.- El usuario elige hora para la ducha
- 17c.- Método POST parámetros->\$idDucha ->reservarCitaDucha.php



```
18c.- execSQL(INSERT INTO Duchan(idUsuario,idDucha,fecha) VALUES
('usu_id','idDucha','fecha'))
19c.- execSQL(SELECT hora FROM Ducha WHERE idDucha='idDucha');
20c.- new
21c.- $query=$this->db->query($sql)->result()
22c.- return $query->result()
23c.- echo 'Tienes una cita para la ducha:' $row["hora"]
```



B.8. Diagrama de Secuencia: Pedir cita para la lavandería

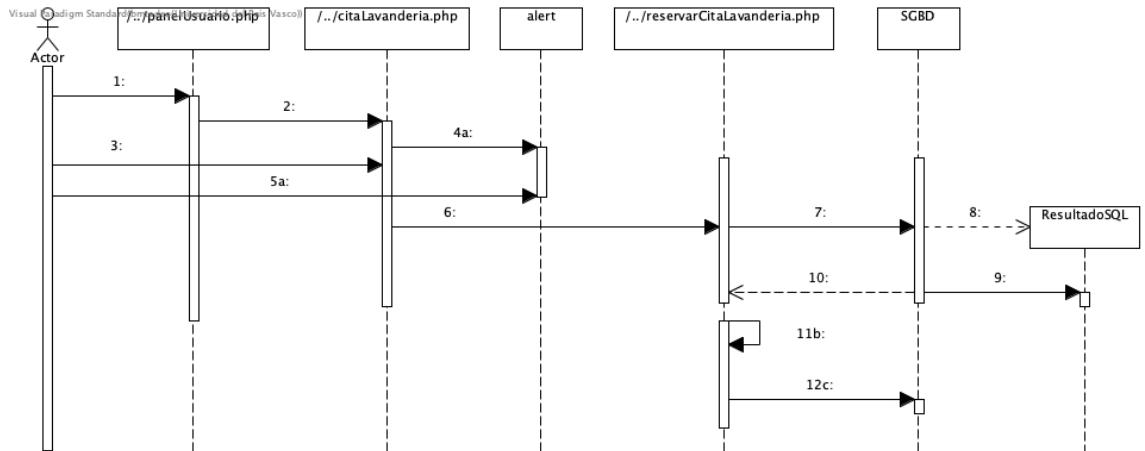


Figura 113: Diagrama de Secuencia: Pedir cita para la lavandería

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Cita Lavandería'.
- 2.- new /.../citaLavanderia.php
- 3.- El usuario selecciona fecha.
(Si la fecha seleccionada es o sábado o domingo)
 - 4a.- new alert('Lo sentimos, nuestras instalaciones no abren los sábados y domingos')
 - 5a.- El usuario pulsa 'Aceptar'
- 6.- Método POST parámetros->\$fecha ->reservarCitaLavanderia.php
- 7.- execSQL(SELECT idUsuario FROM Usuario WHERE '\$usu_id' in (SELECT idUsuario FROM InfoLavanderia WHERE fecha ='\$fecha'))
- 8.- new
- 9.- \$query=\$this->db->query(\$sql)->result()
- 10.- return \$query->result()
(Si \$usu_id!= 'admin' y \$usu_id==\$row['idUsuario'])
- 11b.- echo 'Ya has elegido este dia para entregar la ropa.'
- (Si no)
- 12c.- execSQL(INSERT INTO InfoLavanderia(idUsuario,fecha, entregada) VALUES ('\$usu_id','\$fecha',false))



B.9. Diagrama de Secuencia: Ver citas ducha

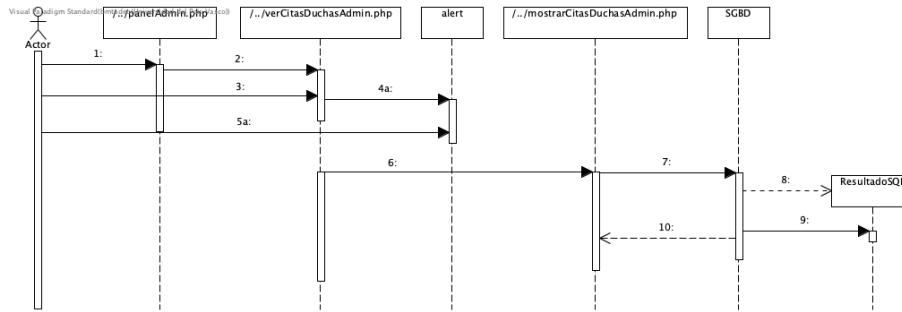


Figura 114: Diagrama de Secuencia: Ver citas ducha

■ Flujo de eventos:

- 1.- El administrador pulsa en 'Ver citas ducha'
- 2.- new ../verCitasDuchasAdmin.php
- 3.- El usuario selecciona fecha.
(Si la fecha seleccionada es o sábado o domingo)
- 4a.- new alert('Lo sentimos, nuestras instalaciones no abren los sábados y domingos')
- 5a.- El usuario pulsa 'Aceptar'
- 6.- Método POST parámetros->\$fecha ->mostrarCitasDuchasAdmin.php
- 7.- execSQL(SELECT Usuario.nick, Usuario.nombre, Usuario.apellido1, Usuario.apellido2, Usuario.qr, Duchan.fecha, Ducha.duchaFisica, Ducha.averiada, Ducha.hora FROM Usuario INNER JOIN Duchan ON Usuario.idUsuario=Duchan.idUsuario INNER JOIN Ducha ON Duchan.idDucha=Ducha.idDucha WHERE Duchan.fecha='\\$fecha');
- 8.- new
- 9.- \$query=\$this->db->query(\$sql)->result()
- 10.- return \$query->result()



B.10. Diagrama de Secuencia: Ver citas lavandería

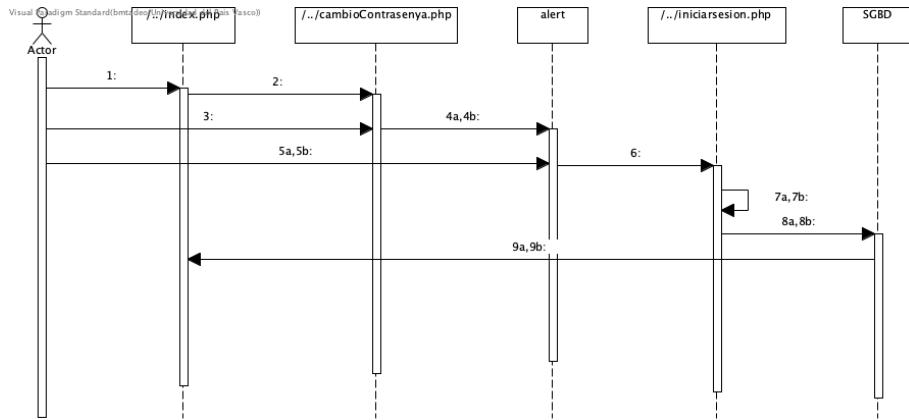


Figura 115: Diagrama de Secuencia: Ver citas lavandería

■ Flujo de eventos:

- 1.- El administrador pulsa en 'Ver citas lavandería'
- 2.- new ../verCitasLavanderiasAdmin.php
- 3.- El usuario selecciona fecha.
(Si la fecha seleccionada es o sábado o domingo)
 - 4a.- new alert('Lo sentimos, nuestras instalaciones no abren los sábados y domingos')
 - 5a.- El usuario pulsa 'Aceptar'
- 6.- Método POST parámetros->\$fecha ->mostrarCitasLavanderiaAdmin.php
- 7.- execSQL(SELECT Usuario.nick, Usuario.nombre, Usuario.apellido1, Usuario.apellido2, Usuario.qr, InfoLavanderia.fecha, InfoLavanderia.entregada FROM Usuario inner join InfoLavanderia ON Usuario.idUsuario=InfoLavanderia.idUsuario WHERE InfoLavanderia.fecha='\$fecha' AND entregada IS FALSE);
- 8.- new
- 9.- \$query=\$this->db->query(\$sql)->result()
- 10.- return \$query->result()
(Si el usuario ha entregado la ropa)
 - 11b.- Método POST parámetros->\$user ->entregarRopaAdmin.php
 - 12b.-execSQL(UPDATE InfoLavanderia SET entregada = true WHERE idUsuario = (SELECT idUsuario FFROM Usuario WHERE nick ='\$user') AND fecha ='\$fecha')
 - 13b.- header("Location: ../mostrarCitasLavanderiaAdmin.php");



B.11. Diagrama de Secuencia: Ver usuarios

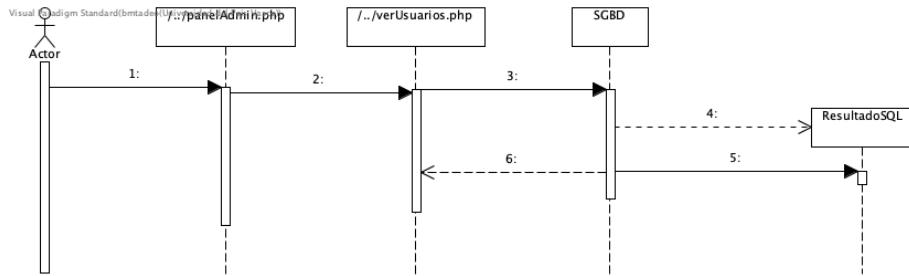


Figura 116: Diagrama de Secuencia: Ver usuarios

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Ver usuarios'
- 2.- new `/../verUsuarios.php`
- 3.- `execSQL(SELECT * FROM Usuario WHERE registroCompleto IS TRUE)`
- 4.- new 5.- `$query=$this->db->query($sql)->result()`
- 6.- `return $query->result()`



B.12. Diagrama de Secuencia: Ver usuarios - Editar Usuario

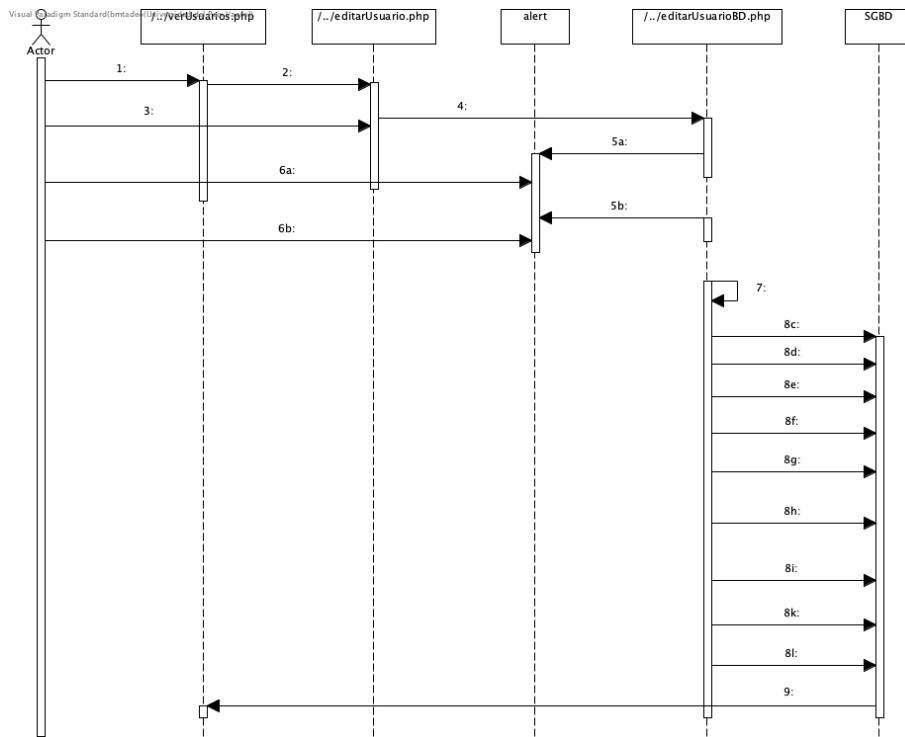


Figura 117: Diagrama de Secuencia: Ver usuarios - Editar Usuario

■ Flujo de eventos:

- 1.- El usuario pulsa editar usuario
- 2.- new `../editarUsuario.php`
- 3.- El usuario inserta los datos que crea necesarios editar y pulsa 'Editar Usuario'
- 4.- Método POST ->parámetros: \$nombre, \$apellido1, \$apellido2, \$nacimiento, \$sexo, \$documentación, \$nacionalidad, \$email,\$telefono,\$contra->`editarUsuarioBD.php`
(Si el nick esta vacio y `$_SESSION['usu_id']`'=='admin')
- 5a.- new alert('Por favor, selecciona un usuario')
- 6a.- El usuario pulsa 'Aceptar'
- (Si el nick no esta vacío y `$_SESSION['usu_id']`'!= 'admin'))
- 5b.- new alert('Por favor, inicia sesión')
- 6b.- El usuario pulsa 'Aceptar'
- 7.-actualizarDatos(\$conexion, \$nick, \$nombre, \$ape1, \$ape2, \$doc, \$nacimiento, \$telefono, \$sexo, \$nacionalidad, \$email, \$contra);
(Si nombre no esta vacío)
- 8c.- execSQL(UPDATE Usuario SET nombre ='\$nombre' WHERE nick=''\$nick')
(Si apellido 1 no esta vacío)



```
8d.- execSQL(UPDATE Usuario SET nombre ='$ape1' WHERE nick=
'$nick')
(Si apellido 2 no esta vacío)
    8e.- execSQL(UPDATE Usuario SET nombre ='$ape2' WHERE nick=
'$nick')
(Si doc no esta vacío)
    8f.- execSQL(UPDATE Usuario SET nombre ='$doc' WHERE nick=
'$nick')
(Si nacimiento no esta vacío)
    8g.- execSQL(UPDATE Usuario SET nombre ='$nacimiento' WHERE
nick= '$nick')
(Si sexo no esta vacío)
    8h.- execSQL(UPDATE Usuario SET nombre ='$sexo' WHERE nick=
'$nick')
(Si telefono no esta vacío)
    8i.- execSQL(UPDATE Usuario SET nombre ='$telefono' WHERE nick=
'$nick')
(Si nacionalidad no esta vacío)
    8j.- execSQL(UPDATE Usuario SET nombre ='$nacionalidad' WHERE
nick= '$nick')
(Si email no esta vacío)
    8k.- execSQL(UPDATE Usuario SET nombre ='$email' WHERE nick=
'$nick')
(Si contra no esta vacío)
    8l.- execSQL(UPDATE Usuario SET nombre ='$contra' WHERE nick=
'$nick')
9.- header("Location: ../../verUsuarios.php");
```



B.13. Diagrama de Secuencia: Ver usuarios - Borrar Usuario

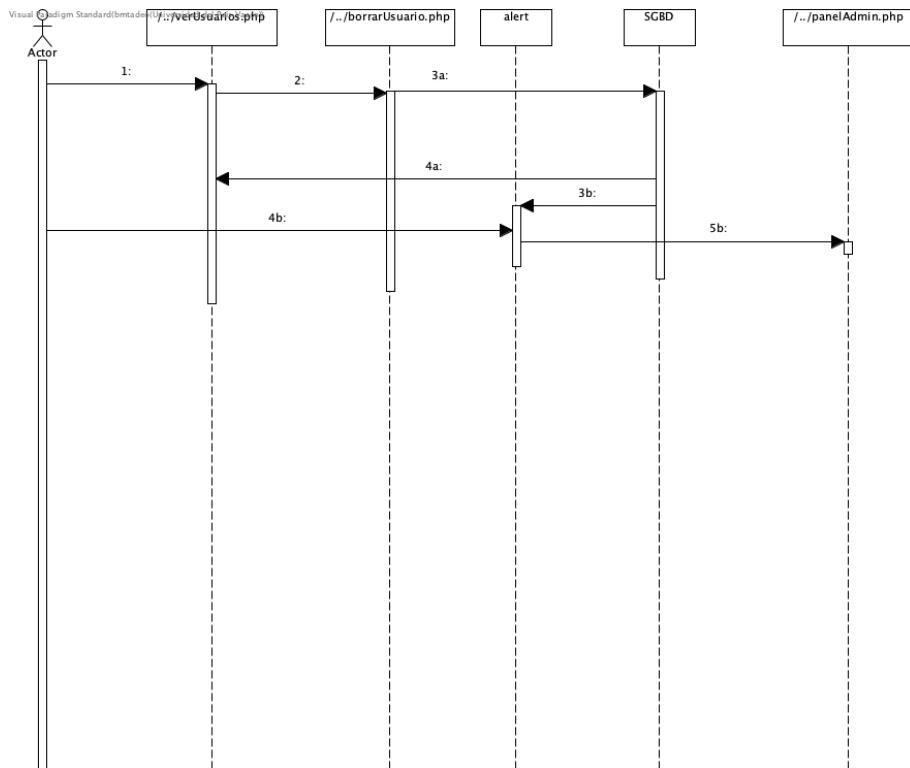


Figura 118: Diagrama de Secuencia: Ver usuarios - Borrar Usuario

■ Flujo de eventos:

- 1.- El usuario selecciona un usuario y pulsa en 'Borrar Usuario'
- 2.- Método POST ->parámetros: \$nick ->borrarUsuario.php
(Si \$_SESSION['usu_id']=='admin'))
 - 3a.- execSQL(DELETE FROM Usuario WHERE nick = '\$nick'))
 - 4a.- header("Location: ..es/verUsuarios.php");
- (Si no)
 - 3b.- new alert('Por favor, inicia sesión')
 - 4b.- El usuario pulsa 'Aceptar'
 - 5b.- header("Location: /..index.php")



B.14. Diagrama de Secuencia: Gestionar clases de castellano - Añadir Clase

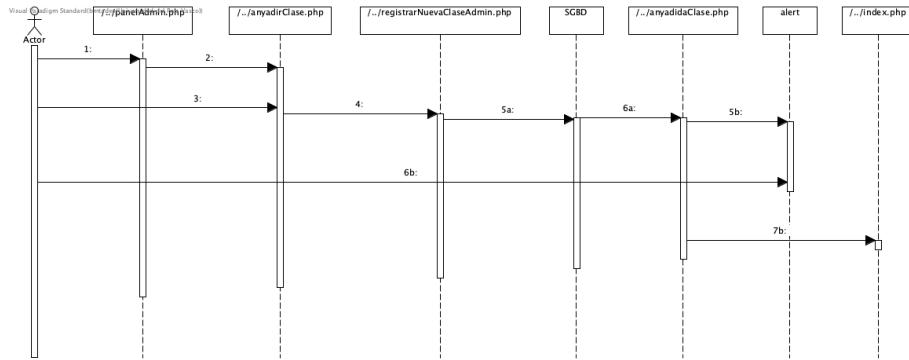


Figura 119: Diagrama de Secuencia: Gestionar clases de castellano - Añadir Clase

■ Flujo de eventos:

- 1.- El usuario selecciona 'Añadir Clase'
- 2.- new ../anyadirClasesAdmin.php
- 3.- El usuario selecciona el dia de la semana de la clase, la hora de la clase, el nivel y la capacidad y pulsa 'Añadir Clase'
- 4.- Método POST ->parámetros: \$fecha, \$hora, \$nivel, \$capacidad ->registrarNuevaClaseAdmin.php
(Si \$_SESSION['usu_id']=='admin')
- 5a.- execSQL(INSERT INTO Clases (nivel, hora, fecha, capacidad) VA-LUES ('\$nivel', '\$hora', '\$fecha', '\$capacidad'))
- 5b.- header("Location: ../anyadidaClase.php")
- 6a.- header("Location: ../index.php");
(Si no)
- 6b.- new alert('Por favor, inicia sesión')
- 6b.- El usuario pulsa 'Aceptar'
- 7b.- header("Location: ../index.php");



B.15. Diagrama de Secuencia: Gestionar clases de castellano - Eliminar Clase

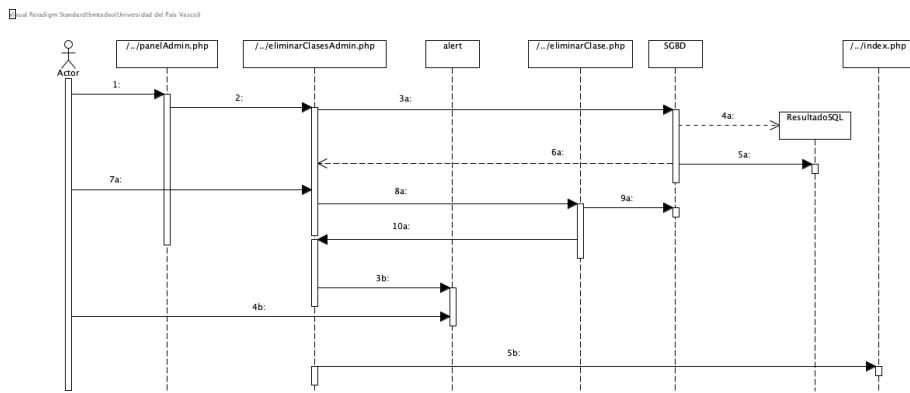


Figura 120: Diagrama de Secuencia: Gestionar clases de castellano - Eliminar Clase

- Flujo de eventos:

- 1.- El usuario selecciona 'Eliminar Clase'
 - 2.- new ../eliminarClasesAdmin.php
(Si \$_SESSION['usu_id']=='admin')
 - 3a.- execSQL(SELECT * FROM Clases;))
 - 4a.- new
 - 5a.- \$query=\$this->db->query(\$sql)->result()
 - 6a.- return \$query->result()
 - 7a.- El usuario selecciona un grupo para eliminar
 - 8a.- Método POST ->parámetros: \$idClase ->eliminarClase.php
 - 9a.- execSQL(DELETE FROM Clases WHERE idClase ='\$id')
 - 10a.- header("Location: ../eliminarClasesAdmin.php");
 - (Si no)
 - 3b.- new alert('Por favor, inicia sesión')
 - 4b.- El usuario pulsa 'Aceptar'
 - 5b.- header("Location: ../../index.php");



B.16. Diagrama de Secuencia: Gestionar clases de castellano - Matricular Alumno

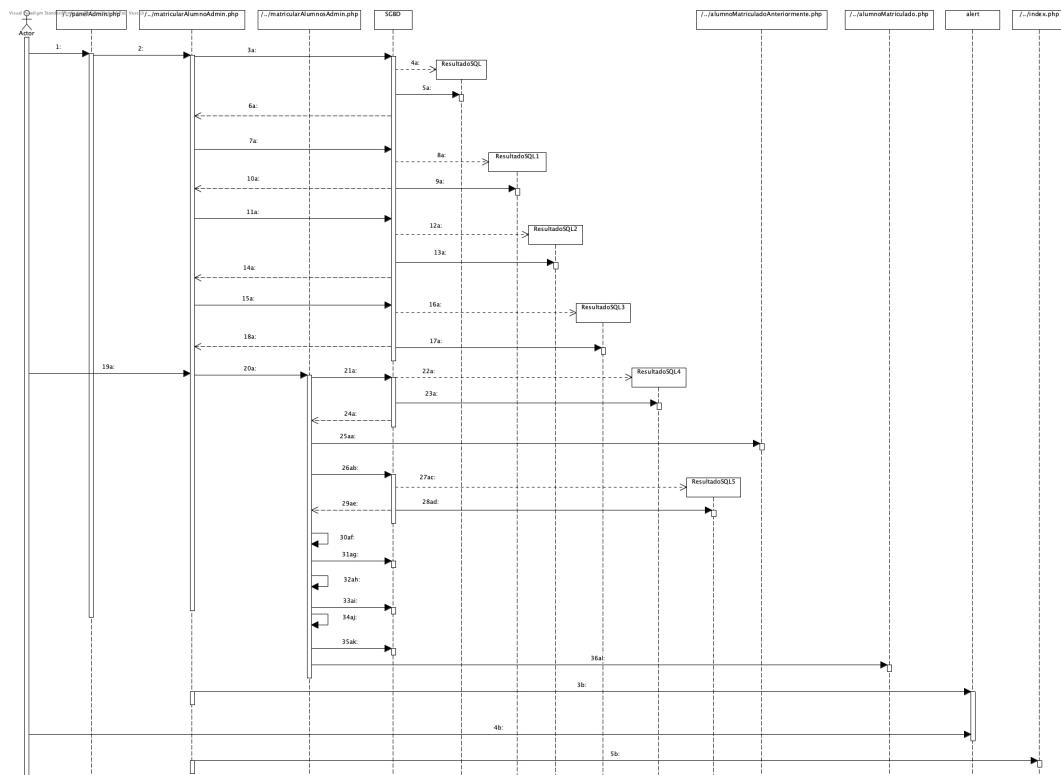


Figura 121: Diagrama de Secuencia: Gestionar clases de castellano - Matricular Alumno

■ Flujo de eventos:

- 1.- El usuario selecciona 'Matricular Alumno'
- 2.- new ../matricularAlumnoAdmin.php
(Si \$_SESSION['usu_id']=='admin')
- 3a.- execSQL(SELECT Usuario.nick, Usuario.nombre, Usuario.apellido1, Usuario.apellido2, Clases.idClase, Clases.fecha, Clases.hora, Clases.nivel, Apuntado.listaEspera FROM Usuario INNER JOIN Apuntado ON Usuario.idUsuario = Apuntado.idUsuario INNER JOIN Clases ON Apuntado.idClase = Clases.idClase;)
- 4a.- new
- 5a.- \$query=\$this->db->query(\$sql)->result()
- 6a.- return \$query->result()
- 7a.- execSQL(SELECT idClase, fecha, hora, nivel, capacidad FROM Clases WHERE capacidad>0;)
- 8a.- new
- 9a.- \$query=\$this->db->query(\$sql)->result()
- 10a.- return \$query->result()
- 11a.- execSQL(SELECT DISTINCT Usuario.nick FROM Usuario INNER JOIN Apuntado ON Usuario.idUsuario=Apuntado.idUsuario INNER JOIN



```
Clases ON Apuntado.idClase = Clases.idClase;
12a.- new
13a.- $query=$this->db->query($sql)->result()
14a.- return $query->result()
15a.- execSQL(SELECT idClase FROM Clases WHERE capacidad>0)
16a.- new
17a.- $query=$this->db->query($sql)->result()
18a.- return $query->result()
19a.- El usuario selecciona usuario, grupo y pulsa en 'Matricular'
20a.- Método POST ->parámetros: $nick, $idClase ->matricularAlumnosAdmin.php
21a.- execSQL(SELECT * FROM Matricula WHERE idUsuario=(SELECT
idUsuario FROM Usuario WHERE nick ='$usuario') AND idClase ='$grupo')
22a.- new
23a.- $query=$this->db->query($sql)->result()
24a.- return $query->result()
(Si $result >0)
    25aa.- header("Location: ../../alumnoMatriculadoAnteriormente.php");
(Si $result <= 0)
    26ab.- execSQL(SELECT idUsuario FROM Usuario WHERE nick
='$usuario')
    27ac.- new
    28ad.- $query=$this->db->query($sql)->result()
    29ae.- return $query->result()
    30af.- insertMatriculado($idUsuario,$grupo,$conexion)
    31ag.- execSQL(INSERT INTO Matricula (idUsuario, idClase, fe-
cha) VALUES ('$idUsuario', '$grupo', CURDATE()))
    32ah.- eliminarApuntados($conexion,$idUsuario, $grupo);
    33ai.- execSQL(DELETE FROM Apuntado WHERE idUsuario =
'$idUsuario')
    34aj.- actualizarCapacidadClases($conexion,$grupo)
    35ak.- execSQL(UPDATE Clases SET capacidad = capacidad - 1
WHERE idClase ='$idGrupo')
    36al.- header("Location: ../../alumnoMatriculado.php");
(Si no)
    3b.- new alert('Por favor, inicia sesión')
    4b.- El usuario pulsa 'Aceptar'
    5b.- header("Location: ../../index.php");
```



B.17. Diagrama de Secuencia: Gestionar clases de castellano - Desmatriricular Alumno

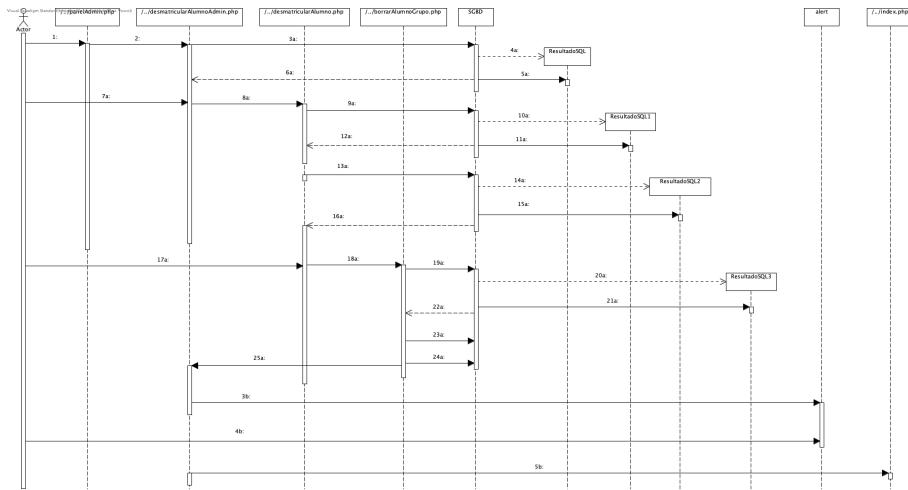


Figura 122: Diagrama de Secuencia: Gestionar clases de castellano - Desmatriricular Alumno

■ Flujo de eventos:

- 1.- El usuario selecciona 'Matricular Alumno'
- 2.- new ../desmatriricularAlumnoAdmin.php
(Si \$_SESSION['usu_id']=='admin')
- 3a.- execSQL(SELECT idClase FROM Clases WHERE capacidad>0;)
- 4a.- new
- 5a.- \$query=\$this->db->query(\$sql)->result()
- 6a.- return \$query->result()
- 7a.- El usuario selecciona grupo y pulsa en 'Mostrar'
- 8a.- Método POST ->parámetros: \$idClase ->desMatriricularAlumno.php
- 9a.- execSQL(SELECT Usuario.nick, Usuario.nombre, Usuario.apellido1, Usuario.apellido2, Clases.idClase FROM Usuario INNER JOIN Matricula ON Usuario.idUsuario = Matricula.idUsuario INNER JOIN Clases ON Matricula.idClase = Clases.idClase WHERE Clases.idClase = '\$idClase';)
- 10a.- new
- 11a.- \$query=\$this->db->query(\$sql)->result()
- 12a.- return \$query->result()
- 13a.- execSQL(SELECT Usuario.nick, Usuario.nombre, Usuario.apellido1, Usuario.apellido2, Clases.idClase, Clases.fecha, Clases.hora, Clases.nivel FROM Usuario INNER JOIN Matricula ON Usuario.idUsuario = Matricula.idUsuario INNER JOIN Clases ON Matricula.idClase = Clases.idClase where Clases.idClase = '\$idClase';)
- 14a.- new
- 15a.- \$query=\$this->db->query(\$sql)->result()
- 16a.- return \$query->result()
- 17a.- El usuario pulsa en 'Eliminar'



```
18a.- Método POST ->parámetros: $nick, $idClase ->borrarAlumnoGrupo.php
19a.- execSQL(SELECT idUsuario FROM Usuario WHERE nick ='$nick')
20a.- new
21a.- $query=$this->db->query($sql)->result()
22a.- return $query->result()
23a.- execSQL(DELETE FROM Matricula WHERE idUsuario ='$idU-
suario' AND idClase ='$idClase')
24a.- execSQL(UPDATE Clases SET capacidad = capacidad + 1 WHE-
RE idClase ='$idClase')
25a.- header("Location: ../es/desmatricularAlumnoAdmin.php")
(Si no)
3b.- new alert('Por favor, inicia sesión')
4b.- El usuario pulsa 'Aceptar'
5b.- header("Location: ../index.php");
```



B.18. Diagrama de Secuencia: Ver ropa

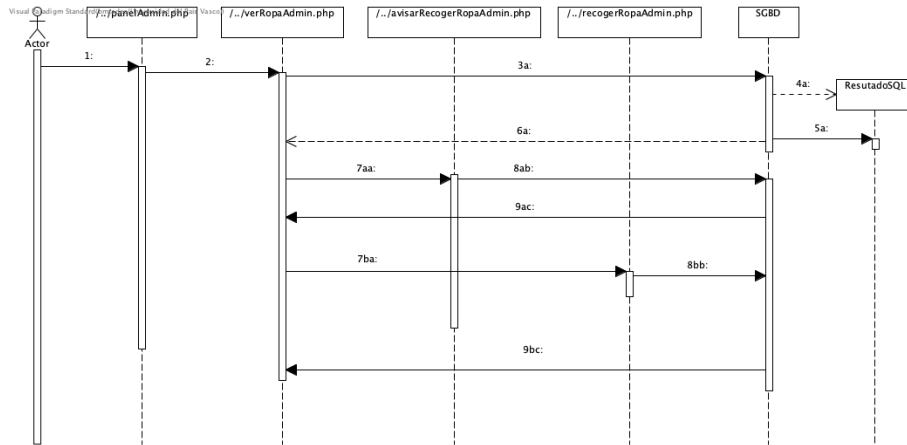


Figura 123: Diagrama de Secuencia: Ver ropa

■ Flujo de eventos:

- 1.- El usuario selecciona 'Ver Ropa Lavanderia'
- 2.- new ../verRopaAdmin.php
(Si \$_SESSION['usu_id']=='admin')
 - 3a.- execSQL(SELECT Usuario.nick, Usuario.nombre, Usuario.apellido1, Usuario.apellido2, Usuario.qr, InfoLavanderia.fecha, InfoLavanderia.entregada, InfoLavanderia.fechaRecogida, InfoLavanderia.recogida FROM Usuario INNER JOIN InfoLavanderia ON Usuario.idUsuario = InfoLavanderia.idUsuario WHERE entregada IS TRUE AND recogida IS FALSE);
 - 4a.- new
 - 5a.- \$query=\$this->db->query(\$sql)->result()
 - 6a.- return \$query->result()
(Si el usuario selecciona el usuario, añade la fecha de recogida y pulsa avisar recogida)
 - 7aa.- Método POST ->parámetros: \$nick, \$fecha ->avisarRecogerRopaAdmin.php
 - 8ab.- execSQL(UPDATE InfoLavanderia SET recogida = true WHERE idUsuario = (SELECT idUsuario FROM Usuario WHERE nick ='\$user') AND entregada IS TRUE)
 - 9ac.- header("Location: ../verRopaAdmin.php")
(Si el usuario selecciona el usuario y pulsa recoger)
 - 7ba.- Método POST ->parámetros: \$nick ->recogerRopaAdmin.php
 - 8bb.- execSQL(UPDATE InfoLavanderia SET recogida = true WHERE idUsuario = (SELECT idUsuario FROM Usuario WHERE nick ='\$user') AND entregada IS TRUE")
 - 9bc.- header("Location: ../verRopaAdmin.php")



B.19. Diagrama de Secuencia: Ver estadísticas del sistema

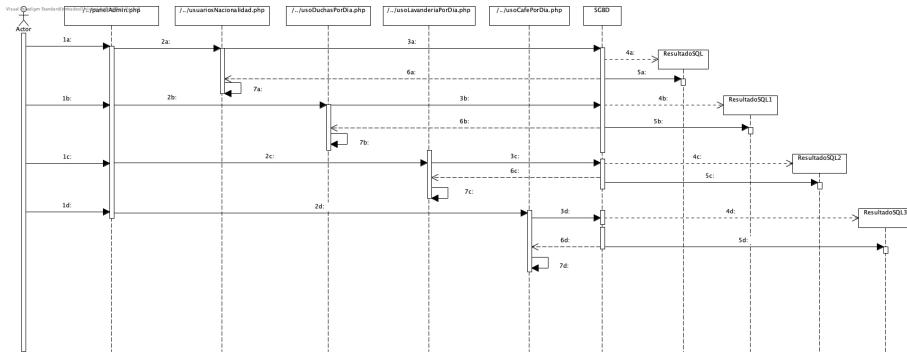


Figura 124: Diagrama de Secuencia: Ver estadísticas del sistema

■ Flujo de eventos:

(Si el usuario elige ver los usuarios por nacionalidad)

1a.- El usuario elige 'Usuarios por nacionalidad'

2a.- new ../usuariosNacionalidad.php

3a.- execSQL(SELECT nacionalidad, count(nacionalidad) FROM Usuario WHERE registroCompleto is true group by nacionalidad;)

4a.- new

5a.- \$query=\$this->db->query(\$sql)->result()

6a.- return \$query->result()

7a.- drawChart()

(Si el usuario elige ver el uso diario de las duchas)

1b.- El usuario elige 'Uso de las duchas'

2b.- new ../usoDuchasPorDia.php

3b.- execSQL(SELECT fecha, count(fecha) FROM Duchan group by fecha;)

4b.- new

5b.- \$query=\$this->db->query(\$sql)->result()

6b.- return \$query->result()

7b.- drawChart()

(Si el usuario elige ver el uso diario de la lavandería)

1c.- El usuario elige 'Uso de la lavandería'

2c.- new ../usoLavanderiaPorDia.php

3c.- execSQL(SELECT fecha, count(fecha) FROM InfoLavanderia group by fecha;)

4c.- new

5c.- \$query=\$this->db->query(\$sql)->result()

6c.- return \$query->result()

7c.- drawChart()

(Si el usuario elige ver el uso diario del servicio del café)

1d.- El usuario elige 'Uso del café'

2d.- new ../usoCafePorDia.php

3d.- execSQL(SELECT fecha, count(fecha) FROM InfoCafe group by fecha;)



```
4d.- new
5d.- $query=$this->db->query($sql)->result()
6d.- return $query->result()
7d.- drawChart();
```



B.20. Diagrama de Secuencia: Apuntarse al café

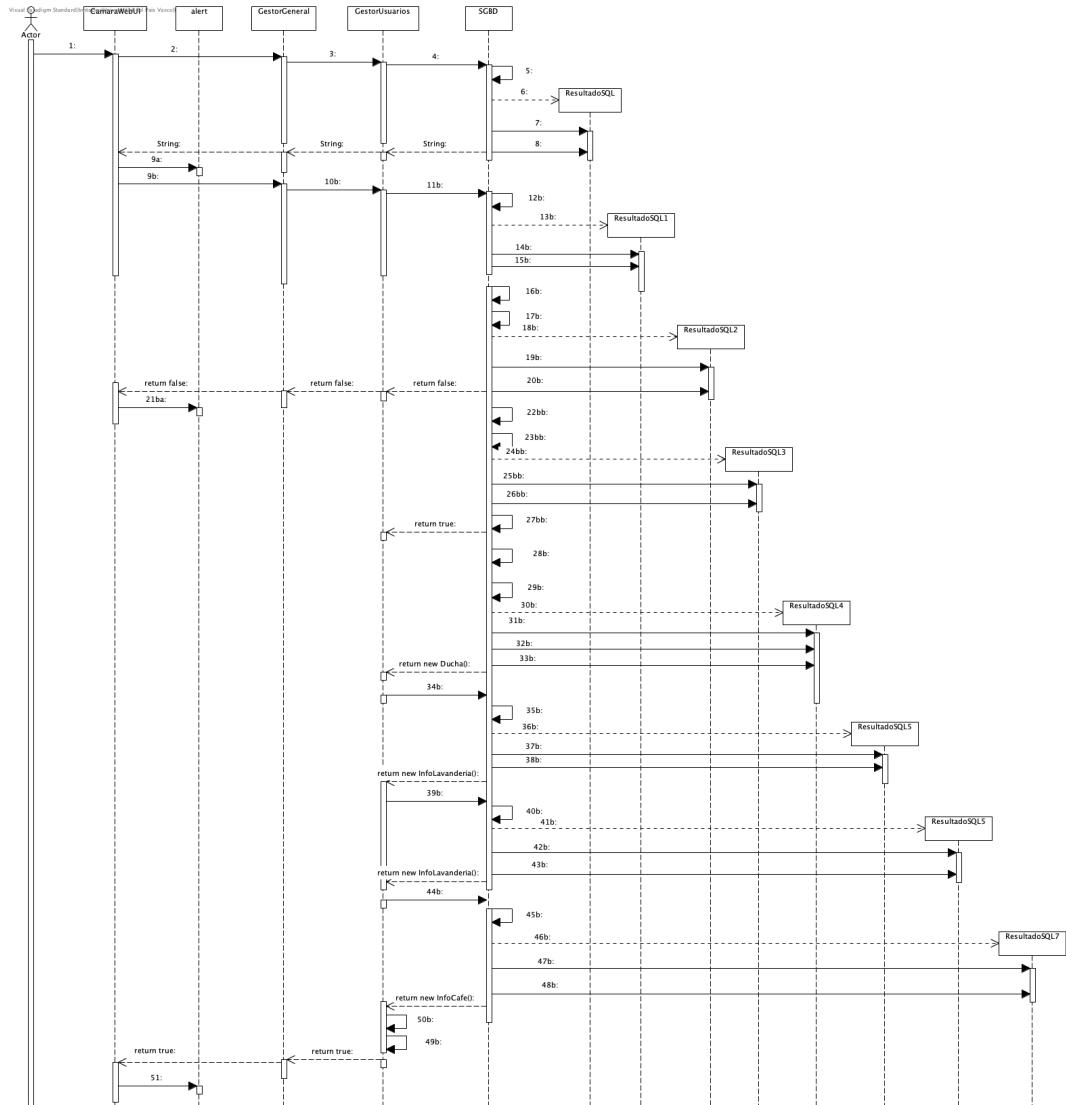


Figura 125: Diagrama de Secuencia: Apuntarse al café

■ Flujo de eventos:

- 1.- El usuario muestra a la Webcam el código QR
- 2.- nombreUsuario(pUsuario): String
- 3.- nombreUsuario(pUsuario): String
- 4.- nombreUsuario(pSQL, pUsuario): String
- 5.- execSQL(SELECT * FROM Usuario WHERE nick = ?)
- 6.- new()
- 7.- next()
- 8.- getString('nombre')



(Si el nombre esta vacio)

9a.- alert(Hola, para usar el servicio hay que registrarse.)

(Si no)

9b.- apuntarAlCafe(pUsuario): boolean

10b.- apuntarAlCafe(pUsuario): boolean

11b.- apuntarAlCafe(pSQL, pUsuario): boolean

12b.- execSQL(SELECT idUsuario FROM Usuario WHERE nick = ?, pUsuario)

13b.- new()

14b.- next()

15b.- getInt('idUsuario')

16b.- buscarUsuarioConTicket(pSQL, fecha, idUsuario): Integer

17b.- execSQL(SELECT idUsuario FROM InfoCafe WHERE fecha= ? AND idUsuario= ?, pUsuario)

18b.- new()

19b.- next()

20b.- getInt('idUsuario')

(Si el usuario ya estaba en la cola)

21ba.- alert(Ya estabas apuntado a la cola.)

(Si el usuario no estaba en la cola)

22bb.- obtenerTurnoAnterior(pSQL, fecha): Integer

23bb.- execSQL(SELECT turno FROM InfoCafe WHERE fecha = ?)

24bb.- new()

25bb.- next()

26bb.- getInt('turno')

27bb.- execSQL(INSERT INTO InfoCafe (idUsuario, fecha, turno) VALUES (?,?,?))

28b.- obtenerTurnoDucha(pSQL, pUsuario): Ducha

29b.- execSQL(SELECT Ducha.idDucha, Ducha.duchaFisica, Ducha.hora FROM Ducha INNER JOIN Duchan ON Ducha.idDucha = Duchan.idDucha WHERE Duchan.idUsuario = ? AND Duchan.fecha = ?)

30b.- new()

31b.- next()

31b.- getInt('idDucha')

32b.- getInt('duchaFisica')

33b.- getString('hora')

34b.- obtenerTurnoLavanderia(pSQL, pUsuario): InfoLavanderia

35b.- execSQL(SELECT InfoLavanderia.fecha FROM InfoLavanderia INNER JOIN Usuario ON InfoLavanderia.idUsuario = Usuario.idUsuario WHERE InfoLavanderia.idUsuario = ? AND InfoLavanderia.fecha = ? AND InfoLavanderia.entregada = ? AND InfoLavanderia.recogida = ?)

36b.- new()

37b.- next()

38b.- getInt('fecha')

39b.- obtenerRopaLavanderia(pSQL, pUsuario): InfoLavanderia

40b.- execSQL(SELECT InfoLavanderia.fechaRecogida FROM InfoLavanderia INNER JOIN Usuario ON InfoLavanderia.idUsuario = Usuario.idUsuario WHERE InfoLavanderia.idUsuario = ? AND InfoLavanderia.entregada = ? AND InfoLavanderia.recogida = ?)



```
41b.- new()
42b.- next()
43b.- getInt('fechaRecogida')
44b.- obtenerTurnoCafe(pSQL, pUsuario): InfoLavanderia
45b.- execSQL(SELECT InfoCafe.turno FROM InfoCafe WHERE Info-
Cafe.idUsuario = ? AND InfoCafe.fecha = ?)
46b.- new()
47b.- next()
48b.- getInt('turno')
49b.- imprimirTicket()
50b.- print()
51.- alert('Recoge el ticket por favor.')
```



B.21. Diagrama de Secuencia: Mantenimiento duchas

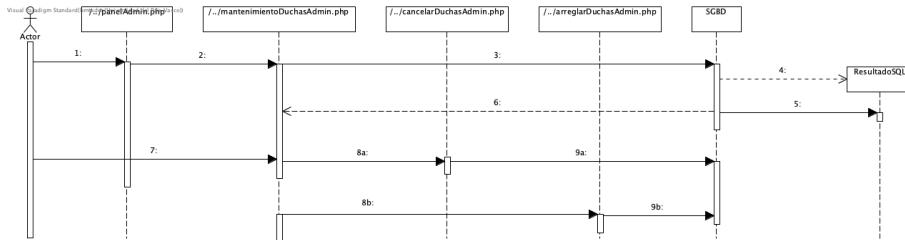


Figura 126: Diagrama de Secuencia: Mantenimiento duchas

■ Flujo de eventos:

- 1.- El usuario selecciona 'Mantenimiento Duchas' y selecciona una fecha
- 2.- new ../mantenimientoDuchasAdmin.php
- 3.- execSQL(Select Ducha.duchaFisica,Ducha.averiada ,Ducha.hora from Ducha inner join Duchan on Ducha.idDucha = Duchan.idDucha where Duchan.fecha='\\$fecha';)
- 4.- new
- 5.- \$query=\$this->db->query(\$sql)->result()
- 6.- return \$query->result()
- 7.- El usuario selecciona
(Si selecciona una ducha y pulsa 'Ducha Averiada')
8a.- Método POST ->parámetros: \$duchaFisica ->cancelarDuchasAdmin.php
9a.- execSQL(UPDATE Ducha SET averiada = TRUE WHERE idDucha = '\$idDucha')
- 8b.- Método POST ->parámetros: \$duchaFisica ->arreglarDuchasAdmin.php
9b.- execSQL(UPDATE Ducha SET averiada = FALSE WHERE idDucha = '\$idDucha')



B.22. Diagrama de Secuencia: Ver reservas ducha

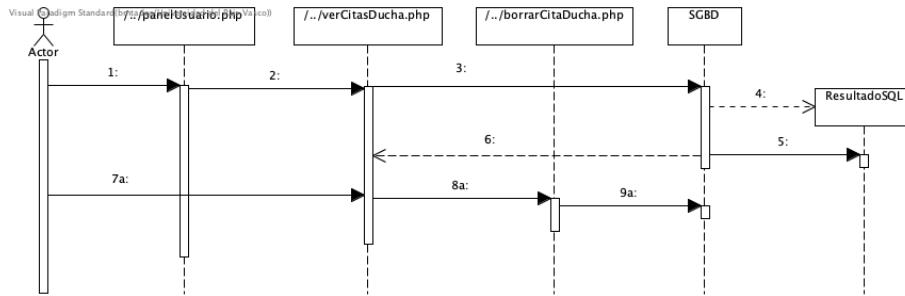


Figura 127: Diagrama de Secuencia: Ver reservas ducha

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Ver citas ducha'
- 2.- new ../verCitasDucha.php
- 3.- execSQL(SELECT Duchan.fecha, Ducha.duchaFisica ,Ducha.avериada, Ducha.hora FROM Duchan INNER JOIN Ducha ON Duchan.idDucha = Ducha.idDucha WHERE Duchan.idUsuario='usu_id' AND Duchan.fecha >= CURDATE() ORDER BY Duchan.fecha;)
- 4.- new
- 5.- \$query=\$this->db->query(\$sql)->result()
- 6.- return \$query->result()
(Si desea una borrar una cita)
- 7a.- El usuario selecciona una fecha de las citas
- 8a.- Método POST ->parámetros: \$fecha ->borrarCitaDucha.php
- 9a.- execSQL(DELETE FROM Duchan WHERE fecha='\$fecha' AND idUsuario = '\$usuario')



B.23. Diagrama de Secuencia: Ver reservas lavandería

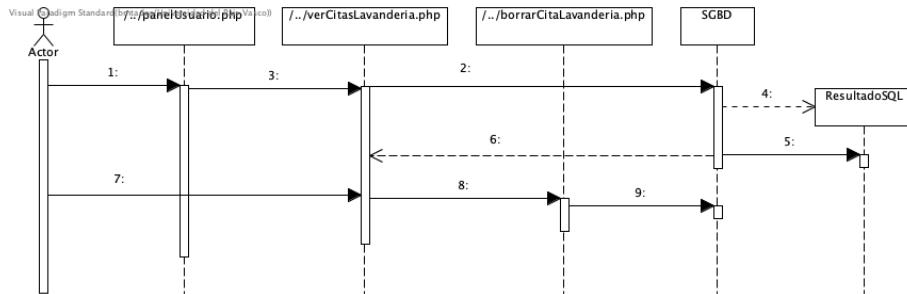


Figura 128: Diagrama de Secuencia: Ver reservas lavandería

■ Flujo de eventos:

- 1.- El usuario pulsa en 'Ver citas lavandería'
- 2.- new ../verCitasLavanderia.php
- 3.- execSQL(SELECT fecha, fechaRecogida, entregada, recogida FROM InfoLavanderia WHERE InfoLavanderia.idUsuario = \$usu_id AND fecha >= CURDATE() AND recogida IS FALSE ORDER BY fecha);
- 4.- new
- 5.- \$query=\$this->db->query(\$sql)->result()
- 6.- return \$query->result()
(Si desea una borrar una cita)
- 7a.- El usuario selecciona una fecha de las citas
- 8a.- Método POST ->parámetros: \$fecha ->borrarCitaLavanderia.php
- 9a.- execSQL(DELETE FROM InfoLavanderia WHERE fecha='<\$fecha' AND idUsuario = '\$usuario')

