

	<b>GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS</b>	<b>CÓDIGO</b>	FO-GS-15
		<b>VERSIÓN</b>	02
<b>ESQUEMA HOJA DE RESUMEN</b>		<b>FECHA</b>	03/04/2017
		<b>PÁGINA</b>	1 de 1
<b>ELABORÓ</b>	<b>REVISÓ</b>	<b>APROBÓ</b>	
Jefe División de Biblioteca	Equipo Operativo de Calidad	Líder de Calidad	

## RESUMEN TRABAJO DE GRADO

AUTOR(ES): NOMBRES Y APELLIDOS COMPLETOS

NOMBRE(S): JHOCEL DUVAN APELLIDOS: SUESCUN TORRES

NOMBRE(S): JAVIER EDUARDO APELLIDOS: CALDERON VILLAMIZAR

FACULTAD: INGENIERÍA

PLAN DE ESTUDIOS: INGENIERÍA DE SISTEMAS

DIRECTOR:

NOMBRE(S): MARCO ANTONIO APELLIDOS: ADARME JAIMES

NOMBRE(S): CARLOS RENÉ APELLIDOS: ANGARITA SANGUINO

TÍTULO DEL TRABAJO (TESIS): IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE LOS CURSOS Y ACTIVIDADES PARA EDUCACIÓN CONTINUA DE LOS PROGRAMAS ACADÉMICOS DE LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER DE LA CIUDAD DE CÚCUTA

El proyecto de grado está centrado en sistematizar la información de los cursos y actividades de educación continua que se lleven a cabo en la Universidad Francisco de Paula Santander de la ciudad de Cúcuta. Los interesados en participar se loguean mediante su correo institucional para realizar su inscripción generando un código QR y así tener control de la asistencia de los participantes mediante lectura de QR a través de un aplicativo móvil. También se generan certificados de participación una vez la educación continua haya terminado, estos certificados son personalizables por el docente encargado. Adicionalmente, el sistema permite la generación de reportes según los formatos establecidos a la fecha por el Sistema Nacional de Información de la Educación Superior (SNIES) cumpliendo con los lineamientos establecidos por el Ministerio de Educación Nacional (MEN).

PALABRAS CLAVES: Educación continua, Software, Código QR, App Móvil, Extreme

Programming

CARACTERÍSTICAS:

PÁGINAS: 277 PLANOS:      ILUSTRACIONES:      CD ROOM:

IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE  
LOS CURSOS Y ACTIVIDADES PARA EDUCACIÓN CONTINUA DE LOS  
PROGRAMAS ACADÉMICOS DE LA UNIVERSIDAD FRANCISCO DE PAULA  
SANTANDER DE LA CIUDAD DE CÚCUTA.

JHOCEL DUVÁN SUESCÚN TORRES  
JAVIER EDUARDO CALDERON VILLAMIZAR

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERÍA  
PLAN DE ESTUDIOS DE INGENIERÍA DE SISTEMAS  
SAN JOSE DE CÚCUTA

2021

IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE  
LOS CURSOS Y ACTIVIDADES PARA EDUCACIÓN CONTINUA DE LOS  
PROGRAMAS ACADÉMICOS DE LA UNIVERSIDAD FRANCISCO DE PAULA  
SANTANDER DE LA CIUDAD DE CÚCUTA.

JHOCEL DUVÁN SUESCÚN TORRES

JAVIER EDUARDO CALDERON VILLAMIEZAR

Trabajo de grado para optar al título de: Ingeniero de Sistemas

DIRECTOR

ING. MARCO ANTONIO ADARME JAIMES

CO - DIRECTOR

ING. CARLOS RENÉ ANGARITA SANGUINO

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA DE SISTEMAS

SAN JOSÉ DE CÚCUTA

2021

**ACTA DE SUSTENTACION DE TRABAJO DE GRADO**

FECHA: 30 de Junio de 2021 HORA: 2:30 PM

Enlace: [meet.google.com/vdb-quns-chg](https://meet.google.com/vdb-quns-chg)

**PLAN DE ESTUDIOS: INGENIERÍA DE SISTEMAS**

**TÍTULO DEL TRABAJO DE GRADO:** "IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE LOS CURSOS Y ACTIVIDADES PARA EDUCACIÓN CONTINUA DE LOS PROGRAMAS ACADÉMICOS DE LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER DE LA CIUDAD DE CÚCUTA."

**JURADOS:**

ING. NELLY ROSANA DIAZ LEAL, ING. MATIAS HERERA CACERES, ING. JUDITH DEL PILAR RODRIGUEZ TENJO

**Director:** Ing. Marco Antonio Adarme Jaines.

**Codirector:** Ing. Carlos Rene Angarita Sanguino

NOMBRE DEL ESTUDIANTE	CÓDIGO	CALIF. NUMERO	CALIF. EN LETRA
JHOCEL DUVÁN SUESCÚN TORRES	1151241	4.5	Cuatro Cinco Cero
JAVIER EDUARDO CALDERÓN VILLAMIZAR	1151229	4.5	Cuatro Cinco Cero

**MERITORIA**

**FIRMA DE LOS JURADOS**

Ing. Nelly Rosana Diaz Leal    Ing. Judith del Pilar Rodríguez Tenjo    Ing. Matías Herrera Cáceres

JUDITH DEL PILAR RODRIGUEZ TENJO  
Coordinadora Comité Curricular

## **Dedicatoria**

A mis padres William e Irgen quienes, con su apoyo incondicional, amor, dedicación y educación lograron inculcar en mi valores, logrando formar la persona quien soy.

A mis tíos Irma e Iván por ser ese apoyo incondicional en mi vida, aconsejándome en las diferentes adversidades que se me presentan.

A mis abuelos José y Rosa, mi otra madre, quienes con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

**Jhocel Duván Suescun Torres**

## **Dedicatoria**

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional.

A mis padres, por ser los pilares más importantes y por haberme formado con buenos sentimientos, hábitos y valores, los cuales me han ayudado a salir adelante en los momentos más difíciles.

A mi hermana, que siempre ha estado junto a mi brindándome su apoyo.

**Javier Eduardo Calderón Villamizar**

## **Agradecimientos**

Agradecemos a Dios por bendecirnos y protegernos durante nuestro camino, ser el apoyo y fortaleza en aquellos momentos de dificultad

Gracias a todas las personas con las que compartimos a lo largo de la carrera universitaria, ya que aportaron directa e indirectamente a nuestro crecimiento personal y profesional.

Agradecemos a nuestros directores de proyecto de grado; Marco Antonio Adarme Jaimés y Carlos René Angarita Sanguino quienes con su apoyo y orientaciones hicieron de este proyecto una realidad.

De igual manera agradecemos a la Universidad Francisco de Paula Santander, a los docentes del Programa de Ingeniería de Sistemas quienes con la enseñanza de sus valiosos conocimientos hicieron que pudiéramos crecer día a día como profesionales, gracias por su paciencia, dedicación y apoyo incondicional.

## **TABLA DE CONTENIDO**

Introducción	24
1. Presentación general del anteproyecto	26
1.1. Título	26
1.2. Planteamiento del problema	26
1.3. Justificación	28
1.4. Objetivos	30
1.4.1. Objetivo general	30
1.4.2. Objetivos específicos	30
1.5. Alcances y delimitaciones	31
1.5.1. Alcance	31
1.5.2. Limitaciones y delimitaciones	31
2. Marco referencial	32
2.1. Antecedentes y estado del arte.	32
2.2. Marco teórico.	37
2.2.1. Quick Response Code (QR Código)	37
2.2.2. Aspectos relacionados a la educación continua	37
2.3. Marco legal.	41
2.3.1. Sistema Nacional de Información de la Educación Superior (SNIES)	41
2.3.2. Acuerdo 082 del 2011 UFPS	41
3. Diseño metodológico	42
4. Cronograma	45
5. Presupuesto	52
6. Análisis de la aplicación ADEEX	54
7. Tecnologías y herramientas computacionales usadas	60
8. Análisis de la información	63
9. Desarrollo del sistema de información	66
10. Estrategia de creación de certificados de participación	109
11. Desarrollo aplicación móvil toma de asistencias	113
12. Migración de información	120
Conclusiones	129
Recomendaciones	130



## LISTADO DE TABLAS

Tabla 1 Aspectos relacionados a la educación continua	38
Tabla 2 Cronograma	45
Tabla 3 Costos de Personal	52
Tabla 4 Costos de Adquisiciones	52
Tabla 5 Costos de Servicios	52
Tabla 6 Costo total del proyecto	53
Tabla 7. Descripción funcionalidades ADEEX	54
Tabla 8. Descripción funcionalidades sistema en curso	57
Tabla 9. Descripción de las tecnologías a utilizar	60
Tabla 10 Historias de Usuario del sistema gestión web	73
Tabla 11 Casos de prueba HU1	75
Tabla 12 Casos de prueba HU2	76
Tabla 13 Casos de prueba HU3	78
Tabla 14 Casos de prueba HU4	80
Tabla 15 Casos de prueba HU5	82
Tabla 16 Casos de prueba HU6	84
Tabla 17 Casos de prueba HU7	85
Tabla 18 Casos de prueba HU8	87
Tabla 19 Casos de prueba HU9	88
Tabla 20 Casos de prueba HU10	89
Tabla 21 Casos de prueba HU11	91
Tabla 22 Casos de prueba HU12	92
Tabla 23 Casos de prueba HU13	93
Tabla 24 Casos de prueba HU15	94
Tabla 25 Casos de prueba HU16	96
Tabla 26 Casos de prueba HU17	98
Tabla 27 Estimación historias de usuario	99

Tabla 28 Plan de iteraciones	100
Tabla 29 Casos de prueba por iteración	100
Tabla 30 Plan de entregas	102
Tabla 31 Historias de usuario Iteración 1	104
Tabla 32 Historias de usuario Iteración 2	104
Tabla 33 Historias de usuario Iteración 3	105
Tabla 34 Historias de usuario Iteración 4	105
Tabla 35 Historias de usuario Iteración 5	106
Tabla 36 Etiquetas plantilla certificados de participación	109
Tabla 37 Historias de usuario aplicación móvil	113
Tabla 38 Casos de prueba HU-A1	114
Tabla 39 Casos de prueba HU-A2	115
Tabla 40 Casos de prueba HU-A3	116
Tabla 41 Estimación historias de usuario aplicación móvil	117
Tabla 42 Plan de iteraciones aplicación móvil	117
Tabla 43 Casos de prueba por iteración aplicación móvil	118
Tabla 44 Historias de usuario Iteración 1 aplicación móvil	118
Tabla 45 Caso de prueba registro de facultad	144
Tabla 46 Caso de prueba registro de facultad ya existente	145
Tabla 47 Caso de prueba actualización de facultad	146
Tabla 48 Caso de prueba registro de departamento académico	147
Tabla 49 Caso de prueba registro de departamento académico ya existente	148
Tabla 50 Caso de prueba actualización de departamento académico	149
Tabla 51 Caso de prueba registro de programa académico	150
Tabla 52 Caso de prueba registro de programa académico ya existente	151
Tabla 53 Caso de prueba actualización de programa académico	152
Tabla 54 Caso de prueba registro exitoso de un usuario en el sistema	157
Tabla 55 Caso de prueba registro fallido de un usuario en el sistema	158
Tabla 56 Caso de prueba registro exitoso de un usuario por parte de otro usuario	159

Tabla 57 Registro fallido de un usuario con número de documento ya existente	161
Tabla 58 Caso de prueba visualizar listado de usuarios	163
Tabla 59 Caso de prueba registro exitoso de curso o evento de educación continua	169
Tabla 60 Caso de prueba registro fallido de curso o evento de educación continua	171
Tabla 61 Caso de prueba visualizar el listado de cursos o eventos de educación continua	173
Tabla 62 Caso de prueba actualización exitosa de curso o evento de educación continua	174
Tabla 63 Caso de prueba inscripción exitosa de un usuario a un curso o evento de educación continua	182
Tabla 64 Caso de prueba visualización de tarjeta de inscripción	183
Tabla 65 Caso de prueba inscripción de un usuario a un curso o evento de educación continua sin cumplir requisitos	184
Tabla 66 Caso de prueba inscripción de un usuario a un curso o evento de educación continua con límite máximo cupos alcanzados	185
Tabla 67 Caso de prueba inscripción de un usuario a un curso o educación continua fuera de fecha límite	186
Tabla 68 Caso de prueba generar reportes SNIES	189
Tabla 69 Caso de prueba visualizar listado de historial de informes SNIES generados	190
Tabla 70 Caso de prueba descarga de informes SNIES generado	191
Tabla 71 Caso de prueba cancelar inscripción de curso o evento de educación continua	193
Tabla 72 Caso de prueba login del sistema con un usuario existente	196
Tabla 73 Login del sistema con usuario nuevo	197
Tabla 74 Caso de prueba registro exitoso de un ponente	206
Tabla 75 Caso de prueba visualizar listado de ponentes registrados	208
Tabla 76 Caso de prueba actualización exitosa de un ponente	209
Tabla 77 Caso de prueba registro exitoso de una jornada	213
Tabla 78 Caso de prueba visualizar listado de jornadas registradas	214
Tabla 79 Caso de prueba actualización exitosa de una jornada	215
Tabla 80 Caso de prueba visualizar listado asistentes	218
Tabla 81 Caso de prueba diseñar diploma de participación	222

Tabla 82 Caso de prueba descarga de diploma de participación por parte del usuario	225
Tabla 83 Caso de prueba marcar asistencia en una jornada	229
Tabla 84 Caso de prueba desmarcar asistencia en una jornada	230
Tabla 85 Caso de prueba asignar permisos a un usuario	233
Tabla 86 Caso de prueba visualizar dashboard con estadísticas	237
Tabla 87 Caso de prueba envío de email de inscripción de curso o evento de educación continua	239
Tabla 88 Caso de prueba envío de email de cancelación de inscripción de curso o evento de educación continua	240
Tabla 89 Caso de prueba envío de email de registro de nuevo usuario en el sistema	241
Tabla 90 Caso de prueba login de app móvil de un usuario con permisos	261
Tabla 91 Caso de prueba login de app móvil de un usuario sin permisos	262
Tabla 92 Caso de prueba login de app móvil de un usuario con permisos	265
Tabla 93 Caso de prueba login de app móvil de un usuario sin permisos	266
Tabla 94 Caso de prueba tomar asistencia a través de la app móvil	270
Tabla 95 Caso de prueba tomar asistencia de un código QR inválido a través de la app móvil	271

## LISTADO DE FIGURAS

Figura 1 Etapas diseño metodológico	42
Figura 2 Ciclo de vida de la metodología XP	67
Figura 3 Actores del sistema web	70
Figura 4 Arquitectura de desarrollo hexagonal	103
Figura 5. Arquitectura despliegue sistema fase desarrollo	107
Figura 6. Arquitectura despliegue sistema fase de producción	108
Figura 7 Interfaz agregar imágenes a certificados de participación	111
Figura 8 Interface agregar textos a certificados de participación	111
Figura 9 Interface agregar líneas a certificados de participación	111
Figura 10 Resultado plantilla certificado de participación	112
Figura 11 Comparación entidad Personas entre ADEEX y GEDUCO	120
Figura 12 Comparación entidad Educación Continua entre ADEEX y GEDUCO	121
Figura 13 Comparación entidad Participantes entre ADEEX y GEDUCO	121
Figura 14 Comparación entidad Jornadas entre ADEEX y GEDUCO	122
Figura 15 Comparación entidad Asistencias entre ADEEX y GEDUCO	122
Figura 16 Comparación entidad Tipos Documento entre ADEEX y GEDUCO	122
Figura 17 Comparación entidad Tipos Educación Continua entre ADEEX y GEDUCO	123
Figura 18 Página 1 de educaciones continuas migradas	124
Figura 20 Página 3 de educaciones continuas migradas	125
Figura 19 Página 2 de educaciones continuas migradas	125
Figura 21 Página 5 de educaciones continuas migradas	126
Figura 22 Página 4 de educaciones continuas migradas	126
Figura 23 Página 6 de educaciones continuas migradas	127
Figura 24 Certificaciones de los cursos realizados por un usuario	127
Figura 25 Diploma de un curso realizado por un usuario	128
Figura 26 Tarjeta Kanban Gestionar áreas académicas.	136
Figura 27 Tarjeta Kanban Gestionar áreas académicas.	136

Figura 28 Modelo de datos áreas académicas	137
Figura 29 Mockup listar facultades	137
Figura 30 Mockup registrar facultad.	138
Figura 31 Mockup listar departamentos	138
Figura 32 Mockup registro departamento	139
Figura 33 Mockup listar programas	139
Figura 34 Mockup registrar programa.	140
Figura 35 Inicialización variables test gestionar facultades.	140
Figura 36 Inicialización variables test gestionar departamentos	141
Figura 37 Resultados test gestionar facultades	141
Figura 38 Test gestionar facultades	141
Figura 39. Resultados test gestionar departamentos	142
Figura 40. Test gestionar departamentos	142
Figura 41 Inicialización variables test gestionar programas	142
Figura 42 Test guardar y buscar director de programa	143
Figura 43. Test listar, buscar programas	143
Figura 44. Resultados test gestionar programas	143
Figura 45 Desarrollo caso de prueba registrar facultad	144
Figura 46 Resultado caso de prueba registro de facultad ya existente.	145
Figura 47 Resultado caso de prueba actualización facultad.	146
Figura 48. Registro caso de prueba registro de departamento académico.	147
Figura 49 Resultado caso de prueba registro de departamento académico ya existente.	148
Figura 50 Resultado caso de prueba actualización de departamento académico	149
Figura 51 Resultado caso de prueba registro de programa académico.	150
Figura 52 Resultado caso de prueba registro de programa académico ya existente.	151
Figura 53 Resultado caso de prueba actualización de programa académico	152
Figura 54. Tarjeta Kanban gestionar usuarios	153
Figura 55 Modelo datos usuarios	154
Figura 56 Mockup listar usuarios	154

Figura 57 Mockup registro usuario.	155
Figura 58 Declaración variables test gestionar usuarios	155
Figura 59 Test guardar diferentes usuarios.	156
Figura 60 Inicialización variables test gestionar usuarios.	156
Figura 61 Resultado caso de prueba registro exitoso de un usuario en el sistema.	158
Figura 62 Resultado caso de prueba registro fallido de un usuario en el sistema.	159
Figura 63 Resultado caso de prueba registro exitoso de un usuario por parte de otro usuario.	161
Figura 64 Resultado caso de prueba registro fallido de un usuario con número de documento ya existente.	162
Figura 65 Resultado caso de prueba visualizar usuarios.	163
Figura 66 Tarjeta Kanban gestionar educaciones continuas.	164
Figura 67 Tarjeta Kanban gestionar educaciones continuas	164
Figura 68 Modelo Entidad-Relación educación continua.	165
Figura 69 Mockup registro educación continua	166
Figura 70 Mockup listar educaciones continuas.	167
Figura 71 Mockup detalles educación continua	167
Figura 72 Instanciación variables gestionar educación continua	168
Figura 73 Test guardar y buscar educación continua	168
Figura 74 Resultados test gestionar educación continua	169
Figura 75 Test eliminar y buscar educación continua por programa	169
Figura 76 Resultado caso de prueba registro exitoso de curso o evento de educación continua.	171
Figura 77 Resultado caso de prueba registro fallido de curso o evento de educación continua	173
Figura 78 Resultado caso de prueba visualizar el listado de cursos o eventos de educación continua.	174
Figura 79 Resultado caso de prueba actualización exitosa de curso o evento de educación continua	177

Figura 80 Tarjeta Kanban realizar inscripción	178
Figura 81 Tarjeta Kanban realizar inscripción	178
Figura 82 Modelo de datos participantes	179
Figura 83 Mockup realizar inscripción	180
Figura 84 Mockup mis participaciones	180
Figura 85. Instanciación variables test realizar inscripción	181
Figura 86 Test realizar inscripción.	181
Figura 87 Resultados test realizar inscripción.	181
Figura 88 Resultado caso de prueba realizar inscripción.	182
Figura 89 Resultado caso de prueba visualización de tarjeta de inscripción.	183
Figura 90 Resultado caso de prueba inscripción fallida a una educación continua sin cumplir requisitos	184
Figura 91 Resultado caso de prueba inscripción fallida por cupo máximo alcanzado	185
Figura 92 Resultado caso de prueba inscripción fallida por fecha límite de inscripción superada.	186
Figura 93 Tarjeta Kanban generar reportes SNIES	187
Figura 94 Modelo Entidad-Relación reportes SNIES	187
Figura 95 Mockup generar reportes SNIES.	188
Figura 96 Test generar reportes SNIES.	188
Figura 97 Resultado test generar reportes SNIES.	189
Figura 98 Resultado caso de prueba generar reportes SNIES.	190
Figura 99 Resultado caso de prueba visualizar reportes SNIES	190
Figura 100 Resultado caso de prueba descarga de reporte SNIES	191
Figura 101 Tarjeta Kanban cancelar inscripción.	192
Figura 102 Mockup cancelar inscripción	192
Figura 103 Test cancelar inscripción.	193
Figura 104 Instanciación variables test cancelar inscripción	193
Figura 105 Resultado caso de prueba cancelar inscripción a una educación continua.	194
Figura 106 Tarjeta Kanban login	195

Figura 107 Mockup inicio sistema	195
Figura 108 Interfaz Login del sistema.	196
Figura 109 Resultado caso de prueba login del sistema con usuario existente.	197
Figura 110 Resultado caso de prueba login del sistema con usuario no existente.	198
Figura 111 Tarjeta Kanban gestionar ponentes.	203
Figura 112 Modelo Entidad-Relación ponentes	204
Figura 113 Mockup listar ponente	204
Figura 114 Mockup registro ponentes	205
Figura 115 Instanciación variables test gestionar ponentes	205
Figura 116 Resultados test gestionar ponentes	206
Figura 117 Test gestionar ponentes	206
Figura 118 Caso de prueba registro exitoso de un ponente.	207
Figura 119 Resultado caso de prueba visualizar listado ponentes	208
Figura 120 Resultado caso de prueba actualización exitosa de un ponente.	209
Figura 121 Tarjeta Kanban gestionar jornadas	210
Figura 122 Modelo Entidad-Relación jornadas	211
Figura 123 Mockup listar jornadas	211
Figura 124 Mockup registro jornada.	212
Figura 125 Instanciación variables test gestionar jornadas	212
Figura 126 Resultados test gestionar jornadas.	213
Figura 127 Test gestionar jornadas	213
Figura 128 Resultado caso de prueba registro exitoso de una jornada	214
Figura 129 Resultado caso de prueba visualización de una jornada	215
Figura 130 Resultado caso de prueba actualización de una jornada.	216
Figura 131 Tarjeta Kanban generar listado asistentes.	217
Figura 132 Modelo Entidad-Relación participantes.	217
Figura 133 Mockup listado participantes.	218
Figura 134 Resultado caso de prueba descargar listado asistentes.	219
Figura 135 Tarjeta Kanban generar certificados de asistencia.	220

Figura 136 Modelo Entidad-Relación certificados	220
Figura 137 Mockup generar certificado diploma, insertar imágenes.	221
Figura 138 Mockup generar certificado diploma, insertar texto.	221
Figura 139 Mockup generar certificado diploma, linea	222
Figura 140 Resultado caso de prueba diseñar diploma de participación.	223
Figura 141 Tarjeta Kanban obtener certificados de asistencia	224
Figura 142 Modelo Entidad-Relación participantes	224
Figura 143 Mockup listado certificaciones.	225
Figura 144 Resultado caso de prueba descarga diploma de participación.	226
Figura 145 Tarjeta Kanban gestionar asistencia	227
Figura 146 Modelo Entidad-Relación asistencias.	227
Figura 147 Mockup toma de asistencias	228
Figura 148 Test toma de asistencias	228
Figura 149 Resultados test toma de asistencias	229
Figura 150 Resultado caso de prueba marcar asistencia de un asistente en una jornada.	230
Figura 151 Resultado caso de prueba desmarcar asistencia de un asistente en una jornada	231
Figura 152 Tarjeta Kanban gestionar roles de usuarios	232
Figura 153 Modelo Entidad-Relación roles usuarios	233
Figura 154 Resultados caso de prueba asignar permisos a un usuario	234
Figura 155 Tarjeta Kanban visualizar datos estadísticos.	235
Figura 156 Mockup datos estadísticos.	235
Figura 157 Instanciación de variables test datos estadísticos.	236
Figura 158 Test datos estadísticos educaciones continuas	236
Figura 159 Resultados test datos estadísticos educaciones continuas.	237
Figura 160 Resultados caso de prueba visualización dashboard estadísticas.	238
Figura 161 Tarjeta Kanban notificación vía email.	239
Figura 162 Resultados caso de prueba envío de email de inscripción de curso o evento de educación continua.	240

Figura 163 Resultado caso de prueba envío de email de cancelación de inscripción de curso o evento de educación continua.	241
Figura 164 Resultado caso de prueba envío de email de registro de nuevo usuario en el sistema.	242
Figura 165 Modelo de datos general	243
Figura 166 Dependencias necesarias para conexión con la base de datos	244
Figura 167 Configuraciones conexión base de datos	244
Figura 168 Bean de configuración para envío de emails	245
Figura 169 Configuración envío de emails	245
Figura 170 Configuración autenticación Google web y App	246
Figura 171 Configuración de seguridad según roles	246
Figura 172 Mapa de navegación de la aplicación web	247
Figura 173 Diagrama UML Persona	247
Figura 174 Diagrama UML jornada y ponente	248
Figura 175 Diagrama UML Educación Continua y Reportes SNIES	248
Figura 176 Diagrama UML inscripción y asistencia	249
Figura 177 Tarjeta Kanban login app móvil	260
Figura 178 Mockup login app	260
Figura 179 Resultado caso de prueba Login de app móvil de un usuario con permisos.	261
Figura 180 Resultado caso de prueba Login de app móvil de un usuario sin permisos	262
Figura 181 Tarjeta Kanban listar educaciones continuas en aplicación móvil	263
Figura 182 Mockup listar educaciones continuas en app	263
Figura 183 Mockup seleccionar jornadas en app	264
Figura 184 Instanciación variables test toma asistencias mediante QR.	264
Figura 185 Resultado test buscar educaciones continuas y jornadas	265
Figura 186 Test buscar educaciones continuas y jornadas	265
Figura 187 Resultado caso de prueba listar las educaciones continuas en la app	266
Figura 188 Resultado caso de prueba listar las jornadas de una educación continua en la app	267

Figura 189 Tarjeta Kanban toma de asistencia mediante QR	268
Figura 190 Mockup tomar asistencia mediante QR	268
Figura 191 Resultado test tomar asistencia app	269
Figura 192 Instanciación variables test toma asistencias mediante QR.	269
Figura 193 Test tomar asistencia app	269
Figura 194 Resultado caso de prueba toma de asistencias a través de la app móvil	270
Figura 195 Resultado caso de prueba toma de asistencia de un código QR inválido a través de la app móvil	271

## RESUMEN

El proyecto de grado está centrado en sistematizar la información de los cursos y actividades de educación continua que se lleven a cabo en la Universidad Francisco de Paula Santander de la ciudad de Cúcuta. Los interesados en participar se loguean mediante su correo institucional para realizar su inscripción generando un código QR y así tener control de la asistencia de los participantes mediante lectura de QR a través de un aplicativo móvil. También se generan certificados de participación una vez la educación continua haya terminado, estos certificados son personalizables por el docente encargado. Adicionalmente, el sistema permite la generación de reportes según los formatos establecidos a la fecha por el Sistema Nacional de Información de la Educación Superior (SNIES) cumpliendo con los lineamientos establecidos por el Ministerio de Educación Nacional (MEN). Para el desarrollo de este sistema se llevó a cabo la metodología Extreme Programming (XP) en cinco fases: exploración, planificación, iteraciones, producción y mantenimiento; permitiendo la extracción de requerimientos, definición de historias de usuario con los respectivos casos de prueba y estableciéndose la arquitectura hexagonal para la construcción del software. Posteriormente se detalla la estrategia utilizada para personalizar los certificados. Por último, se sugieren algunas recomendaciones permitiendo que surjan nuevas investigaciones que contribuyan a ampliar la funcionalidad del sistema.

## ABSTRACT

The degree project is focused on systematizing the information of the continuing education courses and activities that are carried out at the Francisco de Paula Santander University in Cúcuta city. People who are interested in participating have to log in through their institutional email for then registering in some course to generate a QR code and thus have control of the attendance of participants by reading QR through a mobile application. Moreover, the participation certificates are generated once continuing education has ended, these ones are customizable by the teacher in charge. In addition, the information system allows the generation of reports according to the formats established to date by Sistema Nacional de Información de la Educación Superior (SNIES), complying with the guidelines established by the Ministry of National Education (MEN). For the development of this system, the Extreme Programming (XP) methodology was carried out in five phases: exploration, planning, iterations, production and maintenance; achieving the requirements request, user stories definition of with test cases and establishing the hexagonal architecture for the software construction. The strategy used to personalize the certificates is detailed later. Finally, some recommendations are suggested allowing new researches that contributes to expanding the system functionalities.

## **Introducción**

Las Instituciones de Educación Superior (IES) en Colombia se encuentran reguladas por el Ministerio de Educación Nacional (MEN) y este a su vez mediante el Sistema Nacional de Información de la Educación Superior (SNIES) que buscan la planeación, monitoreo, evaluación, asesoría, inspección y vigilancia del sector de la educación. En la actualidad estos procesos se desarrollan utilizando el software ADEEX en el programa de Ingeniería de Sistemas, no obstante los procesos de asistencia y generación de informes son manuales y por eso se replantea la solución para un nuevo software de gestión de educación continua denominado GEDUCO, el objetivo del proyecto de grado es implementar un sistema gestión de actividades de educación continua (cursos, talleres, diplomados, seminarios, congresos y simposios) que se llevan a cabo en los diferentes Programas Académicos de la Universidad Francisco de Paula Santander de la ciudad de Cúcuta (UFPS) con el fin de dar cumplimiento a las políticas y metas de calidad del MEN.

El documento se encuentra estructurado en 12 capítulos de la siguiente forma: En el capítulo 1 se realiza la presentación general del proyecto, donde se presenta el planteamiento del problema, la justificación del proyecto, sus objetivos, sus alcances y delimitaciones. Luego, en el capítulo 2 se presenta el marco referencial, donde se exponen los antecedentes y el marco teórico de los diferentes conceptos manejados. Después, en el capítulo 3 se realiza el diseño metodológico donde se especifican las actividades que se llevaran a cabo para el cumplimiento de los objetivos planteados. Posteriormente, en el

capítulo 4 se define el cronograma con las actividades planeadas en el capítulo anterior. Adicionalmente, en el capítulo 5 se realiza la estimación del presupuesto del proyecto. Mas adelante, en el capítulo 6 se analiza el sistema actual ADEEX que se tiene en el programa de Ingeniería de Sistemas para conocer sus funcionalidades con el fin de planificar lo que se va a incorporar en este proyecto. Seguidamente, en el capítulo 7 se exponen las tecnologías que se utilizaron para desarrollar los productos. Luego, en el capítulo 8 se presenta el análisis de la información según los lineamientos de SNIES para la generación de reportes. En el capítulo 9 se presenta el desarrollo del sistema web explicando la metodología utilizada. Después, en el capítulo 10 se explica la estrategia utilizada para la creación de certificados de participación. Luego, en el capítulo 11 se presenta el desarrollo de la aplicación móvil para la toma de asistencias. Y por último, en el capítulo 12 se expone el proceso de migración de la información existente en ADEEX al nuevo sistema.

El producto final de este proyecto será la implementación de un sistema de información que permita lectura con RFID y QR para la toma de asistencias de los participantes, así como la gestión de informes correspondiente a los lineamientos de SNIES, dicho sistema tendrá el fin de almacenar la información sobre las actividades de educación continua realizadas en la UFPS para que esta pueda hacer su respectivo reporte al SNIES. Dicho sistema web se desarrolló mediante Spring Boot, almacenando sus datos a través del motor de base de datos de MariaDB y utilizando una arquitectura hexagonal. La aplicación móvil se desarrolló para sistemas operativos Android.

## **1. Presentación general del anteproyecto**

### **1.1. Título**

Implementación de un sistema de información para la gestión de los cursos y actividades para educación continua de los Programas Académicos Universidad Francisco de Paula Santander de la ciudad de Cúcuta.

### **1.2. Planteamiento del problema**

Actualmente el Departamento de Sistemas e Informática y el Programa de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander de Cúcuta (UFPS) cuenta con un software web para la administración de los cursos y/o eventos de extensión llamado ADEEX, este software tiene como objetivo un “medio para la toma de decisiones, y recopilar información del Programa Académico de Ingeniería de Sistemas frente a los procesos de autoevaluación, además permite la existencia de unos criterios y políticas institucionales que proporcionan una mayor proyección social, mediante la participación de directivos , profesores y estudiantes del Programa, para generar soluciones a los problemas del entorno mediante proyectos y cursos de extensión acordes a las necesidades regionales y binacionales.” (Gelvez Quintero, 2014).

En el análisis realizado al software, se destacan las siguientes funcionalidades como son: administración de eventos, participantes, lugares, patrocinadores, colaboradores, costos de

inscripción, memorias, ponentes, recursos, informes y diplomas. La toma de asistencia de los eventos se realiza de manera manual, es decir, por medio de una lista impresa se evidencia la asistencia de los participantes en el evento y esto posteriormente es actualizado en el sistema por el docente a cargo, el procesamiento consiste en listas de chequeo manuales cuyo resultado es la generación automática de certificados, sin embargo, estos certificados no pueden ser personalizados. Adicional a esto posee un módulo básico generación de informes.

A pesar de que el Programa de Ingeniería de Sistemas genera informes de sus actividades de educación continua mediante este software para el posterior reporte a SNIES, los demás programas de la UFPS no cuentan con un sistema que permita gestionar estas actividades. Por lo tanto, ellos deben generar dichos informes manualmente y compartírselos a la respectiva dependencia para que estos sean validados y posteriormente enviados al SNIES. Debido a esto no es posible generar reportes estadísticos ni gráficos que proporcionen indicativos según sean los requerimientos de los diferentes Programa Académicos de la UFPS.

De la situación anterior, se deriva que los procesos realizados para la gestión de las actividades de educación continua en los programas de la UFPS son tradicionales y por lo tanto se plantea dar solución a las siguientes preguntas problemáticas: ¿La implementación de una plataforma agilizaría los procesos de gestión en las variables de extensión: cursos y educación continua en la UFPS? ¿El uso de mecanismos hardware y/o dispositivos móviles ayudarían a mejorar la toma de asistencia de los diferentes eventos a realizarse en

la UFPS? ¿La plataforma se podría convertir en una herramienta fiable para indicadores de las variables de extensión: cursos y educación SNIES?

### **1.3. Justificación**

El Sistema Nacional de Información de la Educación Superior (SNIES) ha venido evolucionando tanto en la precisión conceptual como en el alcance de los aspectos que registra sobre la gestión académica, investigativa y de extensión que realizan las Instituciones de Educación Superior (IES). Debido a esto, las IES están en la obligación de suministrar reportes en archivos en Excel con un formato estándar proporcionado a priori que debe ser diligenciado de manera manual o que un sistema lo genere basado estas normas, con el fin que SNIES tome decisiones, diseñe planes y programas que permiten el avance y desarrollo de la educación superior en Colombia. (Ministerio de Educación Nacional, 2017)

En la UFPS los Programas Académicos desarrollan diferentes actividades extracurriculares como cursos y educación continua (cursos, talleres, diplomados, seminarios, congresos, simposios) con el objetivo de compartir nuevos conocimientos a los estudiantes y/o profesionales de la institución.

Por otra parte, la UFPS ha venido implementando el uso de diferentes tecnologías, por ejemplo, la incorporación de RFID en los carnés de los estudiantes y personal académico-administrativo, pese a esto, el carnet solo es utilizado para el acceso al campus universitario. Esta tecnología brinda muchas ventajas como ofrecer gran capacidad de

almacenamiento y velocidad de lectura de datos, lectura de datos a gran distancia, son difíciles de falsificar y ofrecen una trazabilidad individual para cada producto y/o dispositivo. El propósito de este trabajo de grado, será automatizar la asistencia usando esta tecnología para agilizar los procesos de toma de asistencias.

## **1.4. Objetivos**

### **1.4.1. Objetivo general**

Implementar un sistema de información para la gestión de los cursos y actividades para educación continua de los Programas Académicos Universidad Francisco de Paula Santander de la ciudad de Cúcuta.

### **1.4.2. Objetivos específicos**

- Analizar los procesos de ADEEX para generar puntos de convergencia entre esta y la nueva aplicación
- Establecer las tecnologías más apropiadas para la implementación del software
- Desarrollar el sistema web para la gestión de eventos y cursos.
- Migrar la información de ADEEX al sistema propuesto o a implementar

## **1.5. Alcances y delimitaciones**

### **1.5.1. Alcance**

Se desarrollará un sistema de información con el fin de gestionar los cursos y las actividades para la educación continua, tales como: cursos, talleres, diplomados, seminarios, congresos, simposios de la UFPS.

Además, el sistema de información tendrá toma de asistencia mediante el uso de códigos QR generados al momento de realizar la inscripción. Para dicho proceso se contará con una app móvil para dispositivos Android para realizar la lectura de datos.

Inicialmente el sistema será piloto en el Programa de Ingeniería de Sistemas y de igual manera funcional para los demás programas de la UFPS.

### **1.5.2. Limitaciones y delimitaciones**

- Geográfica: La plataforma será para los eventos y cursos de educación continua de la Universidad Francisco de Paula Santander de la ciudad de Cúcuta
- Temporal: Se ejecutará en 8 meses a partir de la aprobación del anteproyecto
- El despliegue queda a responsabilidad de la disposición de la infraestructura de servidores del Programa de Ingeniería de Sistemas
- El sistema no gestionará los procesos internos de aprobación, cronograma y financiación de las variables de extensión de cursos y educación continua

## 2. Marco referencial

### 2.1. Antecedentes y estado del arte.

Para el estudio de antecedentes, se tuvo en cuenta la información presentada en revistas científicas indexadas como IEEE, bases de datos especializadas como ACM0 y buscadores académicos como *google scholar*; tomando una ventana espacio temporal de 5 años. Los términos buscados fueron “QR”, “attendance”, “meetings”, “software web”. Los resultados encontrados se clasificaron a nivel internacional, nacional y local.

A nivel internacional, se han presentado varios proyectos referentes a Códigos QR y RFID, en (Baldeón, Steven, & Cazco, 2019) desarrollaron el proyecto titulado “Sistema informático para automatizar el control de asistencia a clases de docentes y estudiantes de la ESPOCH con dispositivos RFID, aplicando la metodología “SCRUMBAN” donde se desarrolla un software para automatizar el control de asistencia a clases de docentes y estudiantes, se implementó mediante la metodología ágil SCRUMBAN, la cual es una mezcla de SCRUM Y KAMBAN el software está implementada en los lenguajes de programación Java y C++, su despliegue se realiza sobre el servidor de aplicaciones Payara, mientras que para el almacenamiento de los datos se manejó PostgreSQL. Con respecto al hardware se realizó un prototipo de un dispositivo RFID, utilizando arduino MEGA y un lector RFID. Para evaluar la calidad del software, se establecieron dos métricas: tiempo en completar una tarea y proporción productiva; basadas en el estándar de calidad de uso ISO/IEC 9126 4, para lo cual se aplicó el método de captura de tiempos, los datos obtenidos

fueron evaluados mediante la prueba estadística T student, dando como resultado la reducción del tiempo al realizar el proceso de registro de asistencia, mejorando la productividad en la institución.

En (Escobar Chena, 2018) se desarrolló el proyecto titulado “Aplicación web para la gestión de boletos y ticketera móvil para cines”, el proyecto tiene como objetivo implementar una aplicación web para la gestión de boletos y ticketera móvil para el acceso a la sala de cine mediante código QR. También se logró implementar una aplicación móvil para dispositivos Android, una vez que la compra de la entrada ha sido pagada a través de la web, el ticket electrónico es recibido en el teléfono del cliente. Éste contiene en el detalle del ticket un código QR con la información relacionada a la función adquirida que puede ser escaneado por un lector de QR seguidamente, el cliente ingresa un pin de seguridad para confirmar su acceso a la sala de cine, este mecanismo sirve como una llave para el ingreso a la función. De este modo se busca estimular la proactividad del usuario, la eliminación del límite espacio-temporal, la disminución de la intervención humana y la preservación del medio ambiente.

En (Baridura & Nwiabu, 2018) se desarrolló el proyecto titulado “*Developing An Analytic Staff Attendance Management System Using Qr Code*” cuyo objetivo es desarrollar un sistema de gestión de asistencia del personal utilizando el código QR, que resolvería los problemas de asistencia que se enfrentan en las organizaciones, esta herramienta está implementada en Bootstrap y php/MySQL. En el método propuesto, se utiliza el código QR incrustado en una tarjeta para reemplazar el login manual y es escaneado mediante una

cámara web y procesado utilizando Reed – Solomon, corrección de errores hasta que se pueda interpretar adecuadamente la imagen.

En (Robles Ordóñez, 2018) se desarrolló el proyecto titulado “Software para la gestión del Programa de Innovación y Formación Docente de la UTPL” cuyo objetivo es la sistematización de los procesos de control de registro e inscripción, registro de calificaciones, gestión de cursos/componentes, generación de reportes y seguimiento académico de los docentes. Los resultados demuestran que se mejoraron las transacciones y la integridad de los datos teniendo una aceptación de un poco más del 88%.

En (El Mrabet & Moussa, 2017) se desarrolló el proyecto titulado “*Research and design of smart management system in classroom*”, este proyecto propone sistema de gestión de clases para simplificar el seguimiento de la asistencia mediante la identificación del estudiante a través de sistema biométrico, bluetooth, NFC o etiquetas RFID que les notifica a los padres mediante email o SMS. Los resultados demuestran que con el sistema de gestión inteligente y los dispositivos y etiquetas RFID utilizados aumentaron la asistencia de los estudiantes y se redujo el tiempo de envío de información de los estudiantes a sus padres.

En (Tucker, Darcy, & Stantic, 2014) se presenta el proyecto titulado “*A comparative study of RFID technology measuring efficiency and acceptance when capturing attendance*”, este artículo tiene el objetivo de comparar las técnicas y tecnologías (lista manual, RFID y código de barras) de registro de asistencia para determinar el método

óptimo centrado en el precio, la eficiencia y la aceptación del usuario. Los resultados demuestran que la tecnología RFID es la mejor para agilizar la eficiencia y la aceptación del usuario y tomando en cuenta un costo de implementación asequible.

A nivel nacional, se presentan varios proyectos, entre ellos, en (Ñustez Guzman, 2018) se presenta la tesis “Sistema para la integración del proceso de autenticación en aplicaciones web mediante códigos QR” cuyo objetivo es desarrollar un sistema de autenticación basado en códigos QR generados mediante el uso de una biblioteca JavaScript para que posteriormente un usuario pueda leer ese código desde una aplicación móvil también desarrollada, la cual se encarga de entregar la información en tiempo real con solo apuntar la cámara hacia el QR y sin necesidad de digitar información cada vez que se desee ingresar.

También en (Sepúlveda, Yeiny, & Almentero, 2018) se presenta la tesis titulada “Desarrollar una plataforma web para el control de asistencia de estudiantes, docentes y administrativos de la Universidad de Córdoba sede Lorica mediante el uso de tarjetas inteligentes, dispositivos móviles y sms”, con el fin de implementar una herramienta que pueda facilitar el control y registro de asistencias para docentes y personal administrativo mediante tarjetas inteligentes y generar reportes de horarios reduciendo el tiempo y facilitando la organización en la información que se manipula.

A nivel local, se realizó una búsqueda específicamente en el repositorio de la UFPS donde no se encontraron proyectos sobre código QR pero se encontraron dos proyectos que

involucran RFID; uno de ellos es el proyecto “Diseño e Implementación de un sistema automatizado con prototipo RFID controlado por aplicación web y móvil para el acceso vehicular y seguridad del parqueadero de la UFPS utilizando Smart Card” realizado en 2019 y un segundo proyecto titulado “Diseño e implementación de un prototipo con tecnología rfid en un sistema de control por aplicación móvil para el manejo de acceso y seguridad de los laboratorios especializados del departamento de electricidad y electrónica utilizando tarjetas smart card” realizado en 2016.

## **2.2. Marco teórico.**

### **2.2.1. Quick Response Code (QR Código)**

Es un código de matriz 2D que consiste en cuadrados negros dispuestos sobre un fondo blanco (cada uno negro o el módulo blanco representa un 0 o 1), que puede escanearse convenientemente con una aplicación de teléfono inteligente y procesarse utilizando la corrección de errores Reed-Solomon hasta que la imagen se pueda interpretar adecuadamente (Song et al., 2018).

El código QR proporciona una alta capacidad de almacenamiento de datos, escaneo rápido, legibilidad omnidireccional y muchas otras ventajas, incluida la corrección de errores (para que el código dañado también se pueda leer con éxito) y diferentes tipos de versiones (Tiwari, 2016).

### **2.2.2. Aspectos relacionados a la educación continua**

Es la realización de cursos de extensión, actualización o profundización, diplomados y programas de formación docente, articulados con los programas académicos de pregrado y postgrado de la Universidad Francisco de Paula Santander. Asimismo, la realización de eventos como congresos, seminarios, talleres, conferencias, ferias en temas especializados y eventos temáticos. A continuación, en la Tabla 1 se explican los conceptos de los tipos de eventos de educación continua.

Tabla 1 Aspectos relacionados a la educación continua

TIPO EDUCACIÓN	DESCRIPCIÓN
CONTINUA	
Curso	Programa educativo que tiene una duración inferior a 100 horas y que su finalidad es la capacitación, fundamentación y fortalecimiento en diferentes temas propios de la facultad que lo ofrece. Es abierto al público en general y puede ser teórico, práctico o una combinación de ambos
Taller	Programa académico de carácter práctico, donde los participantes adquieren conocimientos y desarrollan habilidades a partir de su propia experiencia. Tiene como finalidad desarrollar temas puntuales encaminados al fortalecimiento de competencias para el trabajo en diversas áreas del conocimiento propias de la facultad que lo ofrece
Seminario	Actividad académica grupal, en la que un número de participantes, profesores o investigadores presentan, analiza y discuten a fondo un tema predeterminado, con el fin de profundizar en un área del conocimiento
Congreso	Evento académico de confrontación entre un grupo de personas calificadas de una misma especialidad, generalmente es un contacto con especialistas a nivel nacional e internacional para el intercambio de experiencias y opiniones, la deliberación de estudios o investigaciones. Las técnicas que se emplean pueden ser: conferencias, talleres de trabajo,

mesas redondas, etc. Al final del evento se elabora una memoria

Simposio	Evento académico que por medio de charlas, discursos o exposiciones minuciosas se presenta a un auditorio un solo tema, con el objetivo de presentar información completa sobre un asunto determinado. Al final debe hacerse un síntesis y trabajo de recolección sobre la documentación en el tema
Diplomado	Programa académico que tiene como propósito la profundización o actualización del conocimiento y/o el desarrollo de competencias y habilidades específicas para el desempeño en el campo laboral o profesional. Se estructura en unidades de enseñanza-aprendizaje auto contenidas a través de módulos sobre un tema determinado, se genera constancia de participación a través de un certificado de aprobación y/o asistencia
Curso de profundización académica	<p>Es un curso relacionado con la naturaleza del Programa Académico, adicional a aquellos del Pensum, orientado a complementar académicamente la formación profesional del estudiante.</p> <p>El curso de profundización tiene una intensidad semestral no menor a 300 horas. Entre 120 y 150 horas se desarrollan de manera presencial estudiante-docente y el tiempo restante corresponderá al trabajo independiente del estudiante</p>

Evento                    Actividad académica de corta duración, dirigida por un grupo de especialistas o “expertos” y de interés común para los participantes, que promueven la reflexión, generación y socialización o consolidación de conocimientos producto del trabajo investigativo, docente y de extensión de la facultad que lo ofrece. Se denominan eventos las conferencias, coloquios, debates, encuentros, exposiciones y similares

---

## **2.3. Marco legal.**

### **2.3.1. Sistema Nacional de Información de la Educación Superior (SNIES)**

Sistema de información que recopila, consolida y organiza información sobre las Instituciones de Educación Superior (IES), sobre la gestión académica que éstas realizan y sobre los programas académicos aprobados por el Ministerio de Educación Nacional, con el fin de producir reportes, estadísticas e indicadores que permiten la planeación, monitoreo, evaluación, asesoría, inspección y vigilancia del sector. El SNIES es la fuente de información oficial del sector de la educación superior en Colombia y como tal tiene la misión de responder a las necesidades de información de todos los usuarios en sus diferentes ámbitos. (Ministerio de Educación Nacional, 2017)

### **2.3.2. Acuerdo 082 del 2011 UFPS**

La UFPS en su compromiso social, liderará proyectos de extensión o proyección social que reflejen una integración con la comunidad, las organizaciones sociales, el Sector Productivo, el Estado, la Academia, las Organizaciones de Cooperación y las instituciones, a través de educación permanente, planes de capacitación institucional, servicios tecnológicos, actividades con egresados, emprendimiento, convenios interinstitucionales, prácticas-pasantías, movilidad estudiantil, investigación y articulación con los entes territoriales, con el fin de contribuir a la transformación social y a la mejora de la calidad de vida de la población

### 3. Diseño metodológico

La implementación del sistema se fundamenta en una investigación aplicada, ya que se aborda un problema en específico cuya solución para lograr los objetivos será el uso de mecanismos o estrategias aplicando varias tecnologías, generando una aplicación web que cumpla con todas las especificaciones solicitadas. La figura 1 presenta las etapas que se llevarán a cabo en el diseño metodológico.

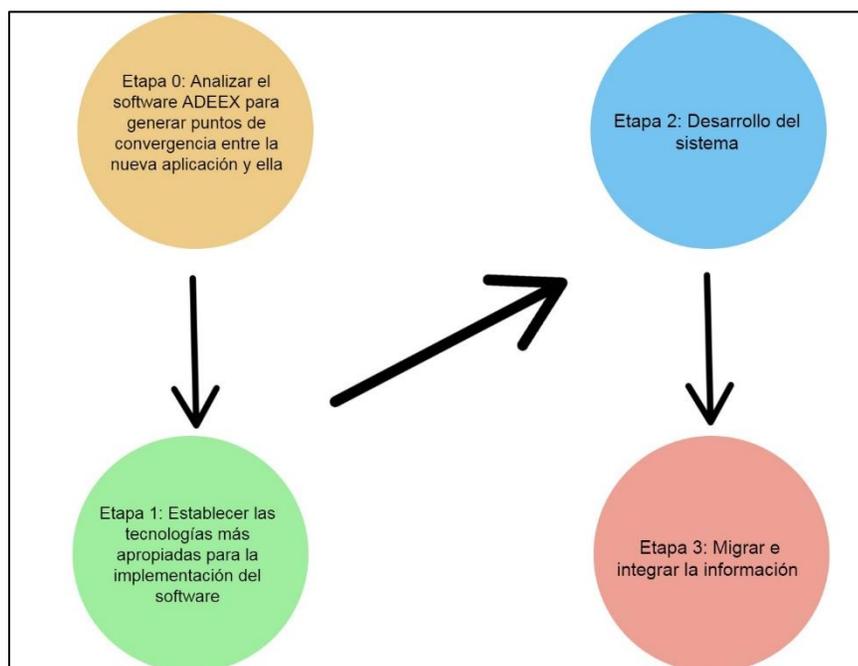


Figura 1 Etapas diseño metodológico

A continuación, se mencionan las actividades que se realizarán para el cumplimiento de los objetivos específicos planteados.

Etapa 0: Analizar el software ADEEX para generar puntos de convergencia entre la nueva aplicación y ella

**Actividades:**

- Identificar las diferentes funcionalidades de ADEEX.
- Identificar los comportamientos de los cursos, talleres, diplomados, seminarios, congresos, simposios establecidos por SNIES

Etapa 1: Establecer las tecnologías más apropiadas para la implementación del software

**Actividades:**

- Identificar las tecnologías que se adapten mejor a la implementación del software
- Definir el hardware a utilizar para la lectura de asistencias.

Etapa 2: Desarrollo del sistema

**Actividades:**

- Definir la metodología de desarrollo de software a utilizar
- Realizar los mockups del sistema a realizar.
- Realizar el modelo de diseño del sistema: diagrama UML del sistema, mapa de navegación y modelo de datos del Sistema.
- Definir la arquitectura del sistema
- Implementar el sistema web.
- Realizar pruebas funcionales, unitarias y de integración del sistema

Etapa 3: Migrar e integrar la información de ADEEX al sistema propuesto o a implementar

**Actividades:**

- Analizar los datos a migrar
- Diseñar el plan de migración
- Realizar la migración de ADDEX al nuevo sistema
- Realizar las pruebas de integridad de datos
- Monitorear el sistema resultante















## 5. Presupuesto

Tabla 3 Costos de Personal

	<b>PARTICIPANTE</b>	<b>ROL</b>	<b>VALOR HORA</b>	<b>HORAS/ SEMANA</b>	<b>HORAS ESTIMADAS</b>	<b>COSTO TOTAL</b>
<b>1</b>	Jhocel Duvan Suescun Torres	Investigador	\$ 25.000	18	432	\$ 10.800.000
<b>2</b>	Javier Eduardo Calderon Villamizar	Investigador	\$25.000	18	432	\$ 10.800.000
<b>3</b>	Ing. Marco Antonio Adarme Jaimes	Director	\$ 55.000	4	96	\$ 5.280.000
<b>4</b>	Ing. Carlos René Angarita Sanguino	Director	\$ 55.000	4	96	\$ 5.280.000
					<b>SUB TOTAL</b>	<b>\$ 32.160.000</b>

Tabla 4 Costos de Adquisiciones

	<b>DESCRIPCIÓN</b>	<b>CANTIDAD</b>	<b>VALOR PROMEDIO</b>	<b>COSTO TOTAL</b>
<b>1</b>	Equipo de cómputo	1	\$ 2.000.000	\$ 2.000.000
<b>2</b>	Libros Digitales	5	\$ 50.000	\$ 250.000
<b>3</b>	Hardware RFID + QR	1	\$350.000	\$350.000
			<b>SUB TOTAL</b>	<b>\$ 2.600.000</b>

Tabla 5 Costos de Servicios

	<b>DESCRIPCIÓN</b>	<b>CANTIDAD</b>	<b>VALOR UNITARIO</b>	<b>COSTO TOTAL</b>
<b>1</b>	Impresiones	1200	\$ 100	\$ 120.000
<b>2</b>	Servicios públicos (luz, Internet,	6	\$ 60.000	\$ 360.000

teléfono)

<b>3</b>	Transporte	645	\$ 1.600	\$ 1.032.000
			<b>SUB TOTAL</b>	<b>\$ 1.512.000</b>

---

Tabla 6 Costo total del proyecto

	DESCRIPCIÓN	VALOR
1	Costos del personal	\$ 32.160.000
2	Costos de adquisiciones	\$ 2.600.000
3	Costos de servicios	\$ 1.512.000
	<b>COSTO TOTAL</b>	<b>\$ 36.272.000</b>

## 6. Análisis de la aplicación ADEEX

En esta etapa se analizó el funcionamiento del aplicativo ADEEX (Gelvez Quintero, 2014) para conocer sus funcionalidades con el fin de planificar lo que se va a incorporar en este proyecto, tomando en cuenta las necesidades que se tienen en la Unidad de Extensión de la UFPS. Se inició revisando el modelo de datos llegando a la conclusión de realizar un refinamiento con el fin de adicionar nuevas funcionalidades. En la Tabla 7 se presenta la descripción de funcionalidades de ADEEX y los roles de usuario que interactúan con ellos.

Tabla 7. Descripción funcionalidades ADEEX

Id	Funcionalidad	Usuario	Funcionamiento
1	Crear eventos	Admin, SuperAdmin	Permite la creación de los diferentes eventos para los cuales se tienen que ingresar unos datos requeridos como: Tipo evento, nombre, fecha inicio, fecha fin, requisitos, fecha límite de inscripción, duración, cantidad máxima de participantes, costo, objetivo, contenido general y un resumen.
2	Visualizar eventos	Admin, SuperAdmin	Permite ver la información de cada evento

3	Actualizar eventos	Admin, SuperAdmin	Permite actualizar la información de los eventos
4	Eliminar eventos	Admin, SuperAdmin	Elimina el evento.
5	Administrar ponentes	Admin, SuperAdmin	Permite agregar, visualizar, actualizar y eliminar los ponentes que se presentarán en los eventos.
6	Administrar tipo de documento	Admin, SuperAdmin	Permite crear, ver, actualizar y eliminar los diferentes tipos de documento permitidos para inscribirse y presentarse en los eventos.
7	Administrar tipo de participante	Admin, SuperAdmin	Permite crear, ver, actualizar y eliminar los tipos de participantes que estarán presentes en los eventos.
8	Administrar tipos de ponencia	Admin, SuperAdmin	Permite crear, visualizar, actualizar y eliminar los tipos de ponencia a presentarse en los eventos.

9	Reportes	Admin, SuperAdmin	Permite generar reportes de diferentes tipos, los cuales son: Reportes de evento de extensión, reporte de participantes de los eventos de extensión, planillas de control de asistencias de los eventos de extensión, reporte de asistencias y/o reporte de inscripciones a los eventos de extensión, reportes de ponentes o docentes de los eventos de extensión, reportes de patrocinadores de los eventos de extensión, reportes de colaboradores de los eventos de extensión, reportes de SNIES, ministerio de educación nacional.  Los reportes que se generan de ADEEX para SNIES están desactualizados.
10	Administración de usuarios	Admin, SuperAdmin	Permite crear, importar, listar, actualizar y eliminar usuarios; además permite administrar los roles de estos usuarios.
11	Realizar inscripción	Usuario final	Permite realizar la inscripción a un evento académico
12	Generar certificados	Usuario final	Permite generar el certificado del evento que participó con una plantilla que no es personalizable. Además, cuando el nombre del evento o del participante es largo al momento de generar certificados estos se desbordan presentando inconformidades con el certificado generado.
13	Actualizar datos	Usuario final	Permite actualizar los datos básicos del usuario final

14	Login	Usuario final	Permite iniciar sesión en el sistema mediante un usuario y contraseña registrado
----	-------	---------------	--

---

A continuación, la Tabla 8 se plantean las nuevas necesidades de mejora o actualización del nuevo sistema, complementando así las existentes en ADEEX e indicando las funcionalidades que se van a realizar en este proyecto.

Tabla 8. Descripción funcionalidades sistema en curso

Id Funcionalidad ADEEX	Id Funcionalidad Sistema En Desarrollo	Mejora y/o actualización
1	1	Se podrán crear eventos con tipos de educación continua dinámicos, es decir, se puede indicar el valor adicional a los ya existentes. Además, se podrán crear tomando como base una ya existente, esto con el fin de mantener un mismo consecutivo interno para efectos del reporte SNIES.
2	2	Se podrán visualizar las educaciones continuas activas o en curso, adicionando un filtro de búsqueda para seleccionar el evento requerido.
3	3	Se podrán actualizar la información de la educación continua.

4	4	No será posible eliminar eventos para tener una trazabilidad en el sistema de los registros que se realicen.
5	5	Se podrán agregar ponentes a la educación continua, adicionando un envío a su correo electrónico registrado de la tarjeta de inscripción que contiene un código QR con el cuál se le tomará la asistencia a dicho evento
9	6	Se tomarán los diferentes formatos previamente definidos por SNIES para generar los reportes de acuerdo con una fecha indicada.
9	7	Se tendrán reportes estadísticos para visualizar gráficamente los datos guardados en el sistema
10	8	Se podrán administrar los diferentes usuarios registrados en el sistema, teniendo en cuenta que es posible la multiplicidad de perfiles, es decir, puede ser estudiante y/o docente y/o administrativo y/o graduado o únicamente externo
10	9	Se podrá otorgar permisos a los diferentes usuarios asociados a la UFPS para que administren las educaciones continuas, administren usuarios y tomen asistencia mediante la aplicación móvil
11	10	El usuario podrá inscribirse a una educación continua, si y solo sí, cumple con alguno de los perfiles (estudiante, docente, administrativo, graduado, externo) a los cuales se ofertó el evento. Dicha inscripción generará una tarjeta de inscripción que sería enviada al correo electrónico del usuario registrado

12	11	Se corregirá el tema del desbordamiento de los diplomas, se utilizará la herramienta CANVAS para evitar que este se desborde cuando el nombre del evento o la persona sea muy largo
N/A	12	La toma de asistencia será mediante una aplicación móvil teniendo en cuenta el QR generado al momento de realizar la inscripción.
13	13	El logueo a la aplicación será mediante Google facilitando el proceso para el usuario.

---

## 7. Tecnologías y herramientas computacionales usadas

A partir de la sección 1.5.1, se estableció el desarrollo de un sistema de información web para la gestión de cursos y eventos de educación continua y una aplicación móvil para la toma de asistencias. La Tabla 9 describe las herramientas que se utilizarán para el desarrollo y producción del sistema.

Tabla 9. Descripción de las tecnologías a utilizar

Herramienta	Versión	Descripción	Uso Dentro Del Proyecto
Spring	2.3.3	Spring es un marco de trabajo, con el cual la programación en Java es más rápida y segura mediante la implementación de varios componentes según sean las necesidades	Para el desarrollo del proyecto, se utilizará Spring Boot como entorno de desarrollo integrado (. Adicionalmente se integrará: -Spring Session para la comunicación con la base de datos. -Spring Data para la manipulación de la información y persistencia con la base de datos. -Spring framework para la inyección de dependencias y gestión de transacciones. -Spring security para el logueo de los usuarios mediante cuenta Google
MySQL	10.4.14	Es un sistema de gestión de base de	Para el desarrollo del proyecto será utilizado para almacenar las entidades con sus respectivas

		datos relacional de relaciones para mantener la integridad de los código abierto datos a gestionar tanto en el sistema web como de la aplicación móvil
Thymeleaf	3.0.11	Es un moderno motor de plantillas Java del lado del servidor para entornos web. Para el desarrollo del proyecto será implementado para el manejo de plantillas HTML que darán a las interfaces de usuario que se manejarán en el sistema
Docker	3.19.03	Es una tecnología de creación de contenedores que permite la creación y el uso de contenedores de Linux para segregar los procesos, de modo que puedan ejecutarse de manera independiente. La tecnología Docker se utilizará para instanciar cada uno de los servicios que utilizaremos como el proxy inverso NGINX, mysql y tomcat
Android Studio	4.1.1	Android Studio es el entorno de desarrollo integrado (IDE) Android. Es el IDE de desarrollo que se utilizará para la construcción de la app móvil para dispositivos Android

---

oficial para el  
desarrollo de apps  
para Android, basado  
en IntelliJ IDEA

---

## 8. Análisis de la información

El Sistema Nacional de Información para la Educación Superior (SNIES) establece en el documento “La Estructura conceptual y funcional” los campos a tener en cuenta para el reporte de información de los cursos y educación continua por parte de las IES. (Ministerio de Educación Nacional, 2017)

### Cursos

La plantilla cursos deberá tener los siguientes campos obligatorios:

- CODIGO\_CURSO: Código del curso
- NOMBRE\_CURSO: Nombre de curso
- ID\_CINE\_CAMPO\_DETALLADO
- ES\_EXTENSION: Es un curso de extensión, digite S o N.
- ACTIVO: Es un curso activo, digite S o N.

### Educación Continua

La plantilla educación continua tendrá almacenada la información de aquellas actividades de formación como cursos y talleres, cuya duración es mínimo 16 horas; los diplomados, cuya duración es mínimo 90 horas; seminarios, congresos y simposios. Esta plantilla tiene subdividida la información en 3 pestañas (EDUCACION\_CONTINUA, EDUCACIÓN\_CONTINUA\_DOCENTES y EDUC\_CONTINUA\_BENEFICIARIOS) adicionales a los siguientes campos

AÑO: año de la educación continua

SEMESTRE: semestre en el cual se llevó a cabo la educación continua

- EDUCACIÓN\_CONTINUA

Los campos obligatorios son:

- CODIGO\_CURSO: Código del curso de la educación continua
  - NUM\_HORAS: Número de horas del curso
  - ID\_TIPO\_CURSO\_EXTENSION: Tipo del curso (cursos, talleres, diplomados, seminarios, congresos, simposios)
  - VALOR\_CURSO: Valor del curso
  - NUM\_ESTUDIANTES: Número de estudiantes del curso
- 
- EDUCACIÓN\_CONTINUA\_DOCENTES
    - ID\_TIPO\_DOCUMENTO: Tipo de documento de identificación (cedula de ciudadanía, documento de identidad extranjera, cedula de extranjería, tarjeta de identidad, pasaporte, certificado cabildo)
    - NUM\_DOCUMENTO: codificación numérica sin puntos, ni comas, ni apostrofes

- EDUC\_CONTINUA\_BENEFICIARIOS
  - ID\_TIPO\_BENEF\_EXTENSION: Tipo de beneficiario (estudiante de las IES, docente de las IES, administrativo de las IES, graduado de la IES, persona no vinculada a la IES)
  - CANTIDAD\_BENEFICIARIOS: Cantidad de beneficiarios

## 9. Desarrollo del sistema de información

### 9.1. Metodología de desarrollo

Para el desarrollo e implementación del sistema se aplicarán los lineamientos de la metodología programación extrema o *eXtremeProgramming* (XP). Esta es una metodología de desarrollo de software basada en los métodos ágiles que evidencia principios tales como el desarrollo incremental, la participación activa del cliente, el interés en las personas y no en los procesos como elemento principal. Esta metodología es muy flexible ante los cambios de requerimientos, sin importar el avance que se tenga del proceso de desarrollo. (Espinosa Conde, 2016)

Además de las características mencionadas anteriormente, la metodología XP por ser una metodología ágil cuenta con muchas ventajas respecto a las metodologías tradicionales como tener procesos más flexibles al permitir subdividir el proyecto en otros más pequeños, sin tener tantas normas o documentación rigurosa que impiden una mejor comunicación con el cliente. (Montero Molina et al., 2018)

Para el caso de estudio, se puede observar que es de vital importancia la participación del cliente y además está sujeto a cualquier cambio de lineamientos presentado por SNIES, por lo cual, debido a las características expuestas anteriormente se establece que la metodología XP es apropiada para el desarrollo del proyecto.

## Fases de la metodología

En la figura 2 se visualiza el ciclo de vida de la metodología a implementar que comprende las diferentes fases a llevar a cabo en la fase de desarrollo de los productos y que serán explicados cada una a continuación.

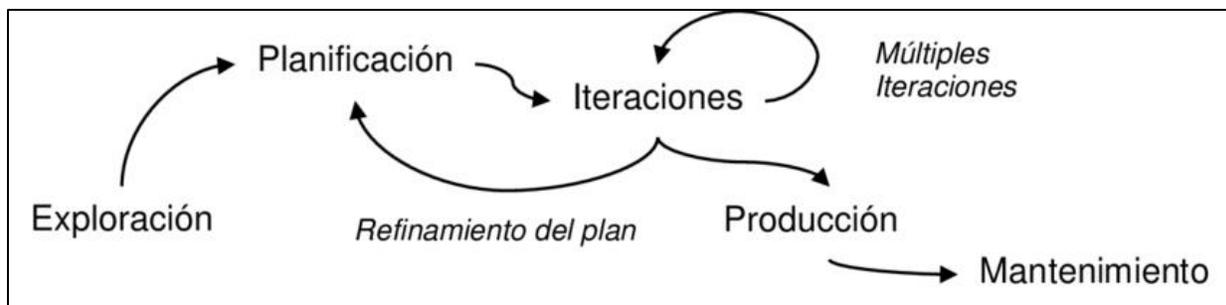


Figura 2 Ciclo de vida de la metodología XP

### 1. Fase de exploración

En esta fase, el cliente define lo que necesita mediante la redacción de sencillas historias de usuarios. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. Al mismo tiempo el equipo de desarrollo se familiariza con las tecnologías que se utilizarán en el proyecto. (Joskowicz, 2008; Letelier & Penadés, 2006)

## **2. Fase de planificación**

El proyecto comienza recopilando historias de usuarios, que sustituyen a los tradicionales casos de uso. Se establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

El resultado de esta fase es un Plan de Entregas o “*Release Plan*”, que es un cronograma de entregas donde se establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de estas. (Joskowicz, 2008)

## **3. Fase de iteraciones**

Esta es la fase principal en el ciclo de desarrollo de XP. Durante esta fase, se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Además, las funcionalidades son desarrolladas, generando al final de cada una, un entregable funcional que implementa las historias de usuario asignadas a la iteración. Estas son también utilizadas para medir el progreso del proyecto, una iteración terminada sin errores es una medida clara de avance. (Joskowicz, 2008)

Debido a que las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se definieron las tareas necesarias de

desarrollo en tarjetas Kanban implementando así un Tablero de Kanban mediante la herramienta web Trello, con esta herramienta las personas asumen ciertas responsabilidades, visualizando los logros y dificultades en el flujo del proceso rápidamente. Además, el equipo se enfoca en terminar las tareas que se tienen asignadas y no acumular tareas iniciadas (Andrés & Torres, 2011).

Para validar que estas historias de usuario se cumplen, XP divide las pruebas del sistema en dos tipos; el primero tipo son las pruebas unitarias o de integración, encargadas de verificar el código y que son diseñadas por los programadores, y el segundo tipo son pruebas de aceptación o pruebas funcionales cuya función es evaluar si al final de cada historia de usuario se consiguió la funcionalidad requerida por el cliente. (Kent, 1999)

#### **4. Fase de puesta en producción**

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste (Joskowicz, 2008).

#### **5. Fase de Mantenimiento**

Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones (Letelier & Penadés, 2006).

### 9.1.1. Fase de Exploración

#### 9.1.1.1. Actores o usuarios del sistema

Un usuario del sistema es cualquier persona que ingresa al sistema mediante logeo para realizar alguna operación (consultar, agregar, actualizar o eliminar) de la información existente. Los usuarios del sistema están acordes a los tipos de beneficiario de cursos y eventos de educación continua establecidos por SNIES como se indicó en el capítulo 6. La figura 3 presenta los usuarios del sistema.

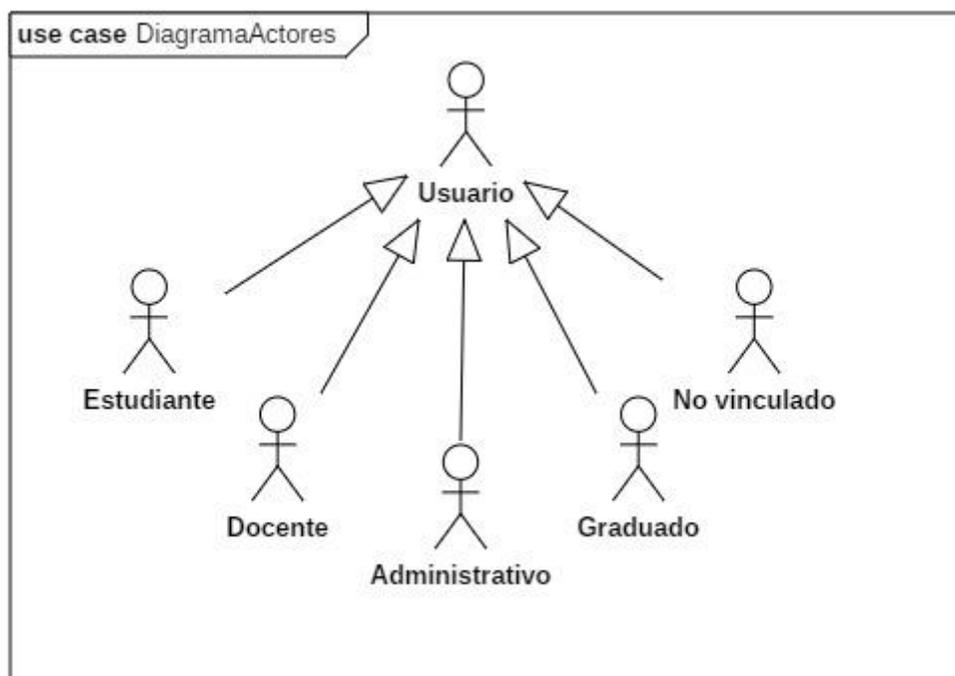


Figura 3 Actores del sistema web

- Estudiante: Es el usuario del sistema perteneciente a un programa académico de la UFPS

- Docente: Es el usuario del sistema adscrito a un departamento académico de la UFPS
- Administrativo: Es el usuario del sistema que labora en algún área administrativa de la UFPS
- Graduado: Es el usuario del sistema que ha finalizado sus estudios en la UFPS
- No vinculado: Es el usuario que no tiene ninguna asociación con la UFPS

#### 9.1.1.2. Roles de usuarios del sistema

Los roles definidos para el sistema de información son los siguientes:

- ROLE\_SUPERADMIN: Es el rol que será el responsable de absolutamente todo el funcionamiento del sistema, teniendo acceso a las diferentes funcionalidades tales como: Gestionar áreas académicas, gestionar educaciones continuas, tomar asistencia mediante la aplicación móvil, inscribirse a educaciones continuas que desee, gestionar personas, generar reportes SNIES y visualizar reportes estadísticos.
- ROLE\_MANAECCU: Es el rol que se otorgará a aquellos usuarios que solo serán responsables de gestionar las educaciones continuas y tomar la asistencia mediante la aplicación móvil.
- ROLE\_MANPEOPLE: Es el rol que se otorgará a aquellos usuarios que solo serán responsables de gestionar la información de los usuarios registrados.

- **ROLE\_ATTENDANCE:** Es el rol que se le otorgará a aquellos usuarios para tomar la asistencia mediante la aplicación móvil, siendo asociados a educaciones continuas en específico.
- **ROLE\_USER:** Es el rol que tendrá cada usuario del sistema por defecto, otorgando la posibilidad de inscribirse a educaciones continuas.
- **ROLE\_SNIES:** Es el rol que se le otorga al director de programa para que genere reportes SNIES de las educaciones continuas de su respectivo programa.

#### **9.1.1.3. Historias de usuario**

Las Historias de usuarios (User stories) sustituyen a los documentos de especificación funcional, y a los casos de uso. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios. Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. (Joskowicz, 2008).

Tomando en cuenta esto, las historias de usuario definidas para el comportamiento del sistema se encuentran establecidas en la Tabla 10:

Tabla 10 Historias de Usuario del sistema gestión web

ID Historia de Usuario	Nombre
HU1	Gestionar eventos y cursos de educación continua
HU2	Gestionar usuarios
HU3	Realizar inscripción
HU4	Generar reportes de SNIES
HU5	Gestionar roles de usuarios
HU6	Gestionar ponentes de la educación continua
HU7	Gestionar jornadas de la educación continua
HU8	Visualizar datos estadísticos
HU9	Generar certificados de participación
HU10	Obtener certificados de participación
HU11	Generar listado de asistentes
HU12	Gestionar asistencia de los asistentes inscritos
HU13	Cancelar Inscripción
HU14	Toma de asistencias mediante QR
HU15	Notificación vía email
HU16	Gestionar las áreas académicas
HU17	Login Sistema

A continuación, se detallarán cada historia de usuario con sus criterios de aceptación y los casos de prueba que se llevarán a cabo.

### **HU1: Gestionar eventos y cursos de educación continua**

Como usuario con rol super-admin, director de programa o docente quiero administrar los cursos y eventos de educación continua para agregar, editar, actualizar los que se presentan en la UFPS.

#### Criterios de aceptación:

\*El usuario con alguno de los roles super-admin, director de programa o docente debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario debe recibir el listado de los cursos y eventos de educación continua registrados según sea el caso del rol:

-Super-Admin: De todos los Programas Académicos.

-Director de Programa: Del Programa Académico a cargo.

-Docente: De los que tiene a cargo.

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder al formulario de registro y diligenciar los campos solicitados correctamente.

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder a las diferentes funcionalidades (editar, ponentes, jornadas, listado participantes, personalizar diploma), en el caso de editar podrá acceder al formulario de registro y actualizar los campos correctamente. En caso de que algún rol quiera editar un registro que no le es visible en el listado, se mostrará un mensaje de error.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder al módulo de educación continua y a cualquiera de las funcionalidades mencionadas anteriormente.

#### Definición casos de pruebas:

A continuación, se definirán en la Tabla 11 los diferentes casos de pruebas de la HU1, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 11 Casos de prueba HU1

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-001	Registro exitoso de curso o evento de educación continua	El usuario registra un curso o evento de educación continua diligenciando el formulario correctamente
CP-002	Registro fallido de curso o evento de educación continua	El usuario registra un curso o evento de educación continua diligenciando el formulario incorrectamente
CP-003	Visualizar el listado de cursos o eventos de educación continua	El usuario visualiza el listado de todos los cursos o eventos de educación continua
CP-004	Actualización exitosa de curso o evento de educación continua	El usuario edita y actualiza los campos de un curso o evento de educación continua diligenciando el formulario correctamente

## HU2: Gestionar usuarios

Como usuario con rol super-admin, admin o director de programa y docente quiero administrar los usuarios del sistema para agregar, editar, actualizar los diferentes usuarios existentes (estudiante, docente, administrativo, graduado, no vinculado) que interactuaran con el sistema.

Criterios de aceptación:

\*El usuario con alguno de los roles super-admin, admin o director de programa debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario debe recibir el listado con los usuarios registrados según sea el caso del rol:

-Super-Admin o admin: De todos.

-Director de Programa: Estudiantes del Programa Académico a cargo.

\*El usuario con alguno de los roles super-admin, admin o director de programa, debe acceder al formulario de registro y diligenciar los campos solicitados correctamente.

\*El usuario con alguno de los roles super-admin, admin o director de programa, debe editar el registro accediendo al formulario y actualizar los campos correctamente. En caso de que algún rol quiera editar un registro que no le es visible en el listado, se mostrará un mensaje de error.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder al módulo de usuario y a cualquiera de las funcionalidades mencionadas anteriormente.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 12 los diferentes casos de pruebas de la HU2, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 12 Casos de prueba HU2

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
----	--------------------	-------------

CP-005	Registro exitoso de un usuario en el sistema	El usuario nuevo se registra diligenciando el formulario correctamente
CP-006	Registro fallido de un usuario en el sistema	El usuario nuevo diligencia el formulario dejando unos campos requeridos sin diligenciar
CP-007	Registro exitoso de un usuario por parte del super admin	El super-admin intenta registrar un usuario diligenciando el formulario correctamente
CP-008	Registro fallido de un usuario con número de documento ya existente	El super-admin intenta registrar un usuario con un número de documento que ya fue registrado
CP-009	Visualizar listado de usuarios	El super-admin visualiza el listado de los usuarios registrados

---

### **HU3: Realizar inscripción**

Como usuario quiero inscribirme a un curso o evento de educación continua para participar en este. los usuarios del sistema para agregar, editar, actualizar los diferentes usuarios existentes (estudiante, docente, administrativo, graduado, no vinculado) que interactuaran con el sistema.

Criterios de aceptación:

- \*El usuario debe estar logeado en el sistema para realizar la inscripción al curso o evento de educación continua
- \*El sistema debe generar al usuario un código QR en una tarjeta de participación que lo identifiquen como participante y/o asistente
- \*El sistema debe impedir al usuario que el participante se inscriba a un curso o evento de educación continua al que ya se encuentra inscrito.
- \*El sistema debe impedir al usuario realizar más inscripciones a un curso o evento de educación continua del límite máximo de participantes registrados.
- \*El sistema debe impedir al usuario realizar inscripciones a un curso o evento de educación continua fuera de la fecha límite registrada
- \*El sistema debe notificar al usuario cualquier error presentado al momento de realizar la inscripción.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 13 los diferentes casos de pruebas de la HU3, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 13 Casos de prueba HU3

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
----	--------------------	-------------

- CP-010    Inscripción exitosa de un usuario a un curso o evento de educación continua disponible de educación continua    El usuario se inscribe a un curso o evento de educación continua disponible de educación continua
- CP-011    Visualización de tarjeta de inscripción del curso o evento de educación continua en los cuales se ha inscrito continua    El usuario consulta la tarjeta de inscripción de los cursos o eventos de educación continua en los cuales se ha inscrito
- CP-012    Inscripción de un usuario a un curso o evento de educación continua sin cumplir con el perfil al cuál se ofertó    El usuario intenta inscribirse a un curso o evento de educación continua activo el cuál fue ofertado a un tipo de público distinto al cuál el usuario tiene sus perfiles
- CP-013    Inscripción de un usuario a un curso o evento de educación continua con límite máximo superado    El usuario intenta inscribirse a un curso o evento de educación continua sin cupos disponibles
- CP-014    Inscripción de un usuario a un curso o evento de educación continua fuera de fecha límite de inscripción    El usuario intenta inscribirse a un curso o evento de educación continua disponible fuera de fecha límite de inscripción
-

#### **HU4: Generar reportes de SNIES**

Como usuario con rol super-admin quiero generar los reportes de cursos y eventos de educación continua en formato SNIES

##### Criterios de aceptación:

\*El usuario con rol super-admin debe estar logueado en el sistema para realizar la generación del reporte

\*El sistema con rol super-admin debe generar al usuario el reporte dado un año en específico y acorde a los formatos establecidos ('Plantilla Cursos.xlsx', 'Plantilla Educación Continua.xlsx' y 'Plantilla Participante') que se encuentran en la url

<https://ww2.ufps.edu.co/vicerrectoria/vicerrectoria-asistente-de-investigacion-y-extension/1589>

\*El sistema con rol super-admin debe permitir al usuario listar el historial de reportes generados por año.

\*El sistema con rol super-admin debe permitir al usuario descargar cualquier reporte generado.

##### Definición casos de pruebas:

A continuación, se definirán en la Tabla 14 los diferentes casos de pruebas de la HU4, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 14 Casos de prueba HU4

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
----	--------------------	-------------

CP-015	Generar reportes SNIES	El usuario selecciona fecha de inicio y fecha fin para generar los reportes de SNIES
CP-016	Visualizar listado de historial de informes SNIES generados	El usuario visualiza el listado de los reportes SNIES (cursos, educación continua y participantes) generados en un rango de fecha en específico
CP-017	Descarga de informe SNIES generado	El usuario descarga el informe de cursos, educación continua y/o participantes generado en formato .xlsx

---

#### **HU5: Gestionar roles de usuarios**

Como usuario con rol super-admin o director de programa quiero asignar o quitar roles a los diferentes usuarios para otorgar o revocar permisos a las funcionalidades del sistema

##### Criterios de aceptación:

El usuario con alguno de los roles super-admin o director de programa debe estar logueado en el sistema para gestionar los roles del usuario

\*El usuario debe acceder al listado de los usuarios registrados según sea el caso del rol:

-Super-Admin: De todos usuarios del sistema.

-Director de Programa: De los estudiantes del Programa Académico a cargo.

\*El sistema debe permitir al usuario con rol super-admin o director del programa cambiar los diferentes roles listados para ese usuario elegido

\*El usuario con alguno de los roles super-admin o director de programa, debe poder cambiar los diferentes roles listados para ese usuario elegido. En caso de que algún rol quiera editar un registro que no le es visible en el listado, se mostrará un mensaje de error.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder cambiar roles de usuario.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 15 los diferentes casos de pruebas de la HU5, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 15 Casos de prueba HU5

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-018	Asignar permisos a un usuario	El usuario otorga/revoca roles a algún usuario

**HU6: Gestionar ponentes de la educación continua**

Como usuario con rol super-admin, director de programa o docente quiero administrar los ponentes de los cursos y eventos de educación continua para consultar, agregar, editar, actualizar y eliminar.

*Criterios de aceptación:*

\*El usuario con alguno de los roles super-admin, director de programa o docente debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder a las diferentes funcionalidades (editar, ponentes, jornadas, listado participantes, personalizar diploma), en el caso de ponentes podrá acceder al listado de ponentes registrados para agregar, editar, actualizar o eliminar según lo desee. En caso de que algún rol quiera consultar los ponentes de un curso o evento de educación continua que no le es visible en el listado de la HU1, se mostrará un mensaje de error.

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder al formulario de registro de ponentes y diligenciar los campos solicitados correctamente.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder al módulo de educación continua y a cualquiera de las funcionalidades mencionadas anteriormente.

*Definición casos de pruebas:*

A continuación, se definirán en la Tabla 16 los diferentes casos de pruebas de la HU6, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 16 Casos de prueba HU6

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-019	Registro exitoso de un ponente	El usuario registra un ponente a un evento o curso de educación continua diligenciando el formulario correctamente
CP-020	Visualizar el listado de ponentes registrados	El usuario visualiza los ponentes registrados en evento o curso de educación continua
CP-021	Actualización exitosa de un ponente	El usuario actualiza la información de un ponente a un evento o curso de educación continua diligenciando el formulario correctamente

### **HU7: Gestionar jornadas de la educación continua**

Como usuario con rol super-admin, director de programa o docente quiero administrar las jornadas de los cursos y eventos de educación continua para consultar, agregar, editar, actualizar y eliminar.

#### *Criterios de aceptación:*

\*El usuario con alguno de los roles super-admin, director de programa o docente debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder a las diferentes funcionalidades (editar, ponentes, jornadas, listado participantes, personalizar diploma), en el caso de jornadas podrá acceder al listado de jornadas registradas para agregar, editar, actualizar o eliminar según lo desee. En caso de que algún rol quiera consultar las jornadas de un curso o evento de educación continua que no le es visible en el listado de la HU1, se mostrará un mensaje de error.

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder al formulario de registro de jornadas y diligenciar los campos solicitados correctamente.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder al módulo de educación continua

Definición casos de pruebas:

A continuación, se definirán en la Tabla 17 los diferentes casos de pruebas de la HU7, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 17 Casos de prueba HU7

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-022	Registro exitoso de una jornada	El usuario registra una jornada a un evento o curso de educación continua diligenciando el formulario correctamente

CP-023	Visualizar listado de jornadas registradas	El usuario visualiza las jornadas registradas en evento o curso de educación continua
CP-024	Actualización exitosa de una jornada	El usuario actualiza la información de una jornada a un evento o curso de educación continua diligenciando el formulario correctamente

---

### **HU8: Visualizar datos estadísticos**

Como usuario con rol super-admin o director de programa o docente quiero visualizar diagramas estadísticos de los cursos y eventos de educación continua.

#### *Criterios de aceptación:*

\*El usuario con alguno de los roles super-admin o director de programa debe estar logueado en el sistema para acceder al módulo de reportes estadísticos.

\*El usuario debe visualizar las gráficas estadísticas de los cursos o eventos de educación continua según sea el caso del rol:

-Super-Admin: De todos los Programas Académicos.

-Director de Programa: Del Programa Académico a cargo.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder al módulo de reportes estadísticos.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 18 los diferentes casos de pruebas de la HU8, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 18 Casos de prueba HU8

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-025	Visualizar dashboard con estadísticas de datos almacenados	El usuario visualiza las estadísticas comparativas de cursos o eventos de educación continua, participantes, asistencias, entre otros

**HU9: Generar certificados de participación**

Como usuario con rol super-admin, director de programa o docente quiero crear y generar el diploma de asistencia a los cursos y eventos de educación continua para los participantes a estos.

Criterios de aceptación:

\*El usuario con alguno de los roles super-admin, director de programa o docente debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder a las diferentes funcionalidades (editar, ponentes, jornadas, listado participantes, personalizar

diploma), en el caso de personalizar diploma podrá acceder al formulario de registro y actualizar los campos correctamente. En caso de que algún rol quiera editar un registro que no le es visible en el listado, se mostrará un mensaje de error.

\*El usuario con alguno de los roles super-admin, admin, director de programa, o docente debe acceder a las diferentes funcionalidades (editar, ponentes, jornadas, listado participantes, personalizar diploma), en el caso de personalizar diploma podrá crear y/o modificar el diploma según lo desee. En caso de que algún rol quiera personalizar el diploma de una educación continua que no le es visible en el listado de la HU1, se mostrará un mensaje de error.

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar a acceder al módulo de educación continua y a cualquiera de las funcionalidades mencionadas anteriormente

Definición casos de pruebas:

A continuación, se definirán en la Tabla 19 los diferentes casos de pruebas de la HU9, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 19 Casos de prueba HU9

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-026	Diseñar diploma de participación	El usuario crea el diploma de asistencia que los participantes descargarán agregándole el fondo,

---

logos, firmas y título/subtítulos  
según sea el caso.

---

### **HU10: Obtener certificados de participación**

Como usuario quiero obtener el certificado de participación de los cursos y eventos de educación continua a los cuales me he inscrito y asistido.

#### Criterios de aceptación:

- \*El usuario debe estar logueado en el sistema para descargar el certificado de participación/asistencia al curso o evento de educación continua.
- \*El usuario debe haber asistido a todas las jornadas del curso o evento de educación continua.
- \*El sistema debe impedir al usuario descargar el certificado de otro participante.

#### Definición casos de pruebas:

A continuación, se definirán en la Tabla 20 los diferentes casos de pruebas de la HU10, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 20 Casos de prueba HU10

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-027	Descarga de diploma de	El usuario descarga su diploma de

---

asistencia por parte del participación en formato .pdf de un  
usuario curso o evento de educación  
continua.

---

### **HU11: Generar listado de asistentes**

Como usuario con rol super-admin, director de programa o docente quiero obtener el listado de las personas inscritas al curso o evento de educación continua.

#### *Criterios de aceptación:*

\*El usuario con alguno de los roles super-admin, director de programa o docente debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder a las diferentes funcionalidades (editar, ponentes, jornadas, listado participantes, personalizar diploma), en el caso de listado de participantes podrá acceder a visualizar los participantes inscritos al curso o evento de educación continua seleccionado. En caso de que algún rol quiera consultar los participantes de un curso o evento de educación continua que no le es visible en el listado de la HU1, se mostrará un mensaje de error.

\*El sistema debe permitirle al usuario generar un archivo en formato pdf con el listado de participantes para ser descargado

\*El usuario con un rol diferente a los mencionados deberá recibir un mensaje de error en caso de intentar acceder al módulo de educación continua y a cualquiera de las funcionalidades mencionadas anteriormente.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 21 los diferentes casos de pruebas de la HU11, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 21 Casos de prueba HU11

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-028	Descarga de listado de asistentes	El usuario descarga el listado de participantes de cualquier curso o evento de educación continua.

**HU12: Gestionar asistencia de los asistentes inscritos**

Como usuario con rol super-admin, director de programa o docente quiero gestionar la asistencia de los participantes inscritos al curso o evento de educación continua.

Criterios de aceptación:

\*El usuario con alguno de los roles super-admin, director de programa o docente debe estar logueado en el sistema para acceder al módulo de educación continua

\*El usuario con alguno de los roles super-admin, director de programa, o docente debe acceder al listado de participantes (H11) y checkear o no la asistencia de un participante en una jornada determinada.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 22 los diferentes casos de pruebas de la HU12, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 22 Casos de prueba HU12

ID	NOMBRE PRUEBA	CASO	DESCRIPCIÓN
CP-029	Marcar asistencia de un asistente en una jornada	un	El usuario checkea asistencia de un participante en una jornada del curso o evento de educación continua.
CP-030	Desmarcar asistencia de un asistente en una jornada	un	El usuario retira asistencia de un participante en una jornada del curso o evento de educación continua.

**HU13: Cancelar Inscripción**

Como usuario quiero cancelar mi inscripción en un curso o evento de educación continua.

Criterios de aceptación:

\*El usuario debe estar logueado en el sistema para cancelar la inscripción al curso o evento de educación continua

\*El sistema debe permitir al usuario cancelar la inscripción al curso o evento de educación continua, si y solo sí, este no ha iniciado.

\*El sistema debe impedir al usuario cancelar la inscripción de otras personas.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 23 los diferentes casos de pruebas de la HU13, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 23 Casos de prueba HU13

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-031	Cancelar inscripción de curso o evento de educación continua	El usuario cancela su inscripción a un curso o evento de educación continua jornada del curso o evento de educación continua.

#### **HU14: Toma de asistencias mediante QR**

Más adelante, en el capítulo 11 se detalla el proceso que se llevó a cabo para la construcción de la aplicación móvil para realizar la toma de asistencias mediante el código QR.

#### **HU15: Notificación vía email**

Como usuario quiero recibir un email a mi correo notificando la inscripción o cancelación de inscripción a un curso o evento de educación continua.

##### Criterios de aceptación:

\*El usuario debe recibir en el email generado la tarjeta de inscripción al curso o evento de educación continua.

\*El usuario debe recibir un email de cancelación de inscripción de un curso o evento de educación continua.

##### Definición casos de pruebas:

A continuación, se definirán en la Tabla 24 los diferentes casos de pruebas de la HU15, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 24 Casos de prueba HU15

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
----	--------------------	-------------

CP-036	Envío de email de inscripción de curso o evento de educación continua	El usuario recibe un email, en el correo de que registró inicialmente, una notificación de inscripción a un curso o evento de educación continua
CP-037	Envío de email de cancelación de inscripción de curso o evento de educación continua	El usuario recibe un email, en el correo de que registró inicialmente, una notificación de cancelación de inscripción a un curso o evento de educación continua.
CP-038	Envío de email de registro de nuevo usuario en el sistema	El usuario recibe un email, en el correo de que registró, una notificación de registro exitoso al sistema.

---

### **HU16: Gestionar las áreas académicas**

Como usuario con rol super-admin quiero administrar las facultades, departamentos y programas académicos de la UFPS

#### *Criterios de aceptación:*

\*El usuario con rol super-admin debe loguearse al aplicativo web mediante Login Google para acceder al sistema.

\*El sistema debe permitirle al usuario con rol super-admin listar, agregar, actualizar las facultades, departamentos y programas académicos.

\*El sistema debe permitirle al usuario con rol super-admin agregar a una facultad varios departamentos y programas académicos.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 25 los diferentes casos de pruebas de la HU16, que se realizarán en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 25 Casos de prueba HU16

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-039	Registro de facultad	El super-admin registra una facultad de la UFPS.
CP-040	Registro de facultad ya existente	El super-admin registra una facultad que ya se registró antes
CP-041	Actualización de facultad	El super-admin actualiza el nombre de una facultad
CP-042	Registro de departamento académico	El super-admin registra un departamento académico de la UFPS.

CP-043	Registro de departamento académico ya existente	El super-admin registra un departamento académico que ya se registró antes
CP-044	Actualización de departamento académico	El super-admin actualiza los datos de un departamento académico correctamente.
CP-045	Registro de programa académico	El super-admin registra un programa académico de la UFPS.
CP-046	Registro de programa académico ya existente	El super-admin registra un programa académico que ya se registró antes
CP-047	Actualización de programa académico	El super-admin actualiza los datos de un programa académico correctamente

---

### **HU17: Login del Sistema**

Como usuario quiero loguearme mediante autenticación Google para acceder a las diferentes funcionalidades del sistema.

#### *Criterios de aceptación:*

\*El usuario debe loguearse al aplicativo web mediante Login Google para acceder al sistema

\*El usuario debe registrarse si no hay datos asociados al correo con el cuál se loguea. Esto solo debe ser posible para usuarios: estudiantes, graduados y no vinculados.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 26 los casos de pruebas de la HU17, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 26 Casos de prueba HU17

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-048	Login del sistema de información con usuario existente	El usuario intenta loguearse mediante autenticación Google.
CP-049	Login del sistema de información con usuario nuevo	El usuario intenta entrar al sistema mediante autenticación Google.

## 9.1.2. Fase de Planificación

### 9.1.2.1. Estimación Historias de usuario

Las historias de usuario se estiman de acuerdo con el grado de prioridad (alto, medio o bajo) otorgado por el cliente, en este caso la persona encargada de la unidad de extensión de la UFPS. Además, el equipo desarrollador estima el esfuerzo de cada historia de usuario con valores de 1 a 3 puntos, siendo el punto la unidad de medida, donde 1 punto equivale a una semana ideal de programación. La Tabla 27 presenta las historias de usuario que se realizarán en el desarrollo del proyecto y su respectiva estimación de esfuerzo.

Tabla 27 Estimación historias de usuario

Historia de Usuario	Nombre	Prioridad	Esfuerzo
HU1	Gestionar eventos y cursos de educación continua	Alta	2
HU2	Gestionar usuarios	Alta	2
HU3	Realizar inscripción	Alta	1
HU4	Generar reportes de SNIES	Alta	2
HU5	Gestionar roles de usuarios	Baja	1
HU6	Gestionar ponentes de la educación continua	Baja	1
HU7	Gestionar jornadas de la educación continua	Baja	1
HU8	Visualizar datos estadísticos	Media	2
HU9	Generar certificados de participación	Baja	2
HU10	Obtener certificados de participación	Baja	2
HU11	Generar listado de asistentes	Alta	1
HU12	Gestionar asistencia de los asistentes inscritos	Media	2
HU13	Cancelar Inscripción	Alta	1
HU14	Toma de asistencias mediante QR	Media	3
HU15	Notificación vía email	Baja	1
HU16	Gestionar las áreas académicas	Alta	1
HU17	Login Sistema	Media	1

### 9.1.2.2. Plan de iteraciones

En este plan, se seleccionan las historias de usuarios, de acuerdo a la estimación realizada en la Tabla 9, para cada entrega y son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Cada historia de usuario se traduce en tareas específicas de programación. De igual manera, para cada historia de usuario se establecen las pruebas de aceptación. (Joskowicz, 2008). En la Tabla 28 se encuentra el plan de iteraciones, donde se indica que historias de usuario se realizarán por cada una de las iteraciones.

Tabla 28 Plan de iteraciones

Iteración	Historias de Usuario
Iteración 1	HU1, HU2, HU3, HU16
Iteración 2	HU4, HU13, HU17
Iteración 3	HU6, HU7, HU11
Iteración 4	HU9, HU10, HU12
Iteración 5	HU5, HU8, HU14, HU15

La ejecución de pruebas se llevará a cabo dependiendo cada iteración como se indicará en la siguiente tabla, donde se puede visualizar que la iteración 1 tiene la mayor cantidad de pruebas porque lo que se requiere un mayor esfuerzo debido a que en esta iteración se encuentran historias de usuario con una prioridad alta y que son fundamentales y base para el desarrollo posterior de las siguientes iteraciones. En la Tabla 29 se presenta los casos de pruebas que se mencionaron anteriormente y que se realizarán por cada una de las iteraciones de desarrollo.

Tabla 29 Casos de prueba por iteración

	ITERACIÓN 1	ITERACIÓN 2	ITERACIÓN 3	ITERACIÓN 4	ITERACIÓN 5
ID	CP-001, CP-	CP-015, CP-	CP-019, CP-	CP-026, CP-	CP-018, CP-
PRUEBA	002, CP-003,	016, CP-017,	020, CP-021,	027, CP-029,	025, CP-032,
	CP-004, CP-	CP-031, CP-	CP-022, CP-	CP-030	CP-033, CP-
	005, CP-006,	048, CP-049	023, CP-024,		034, CP-035,
	CP-007, CP-		CP-028		CP-036, CP-

---

	008, CP-009,				037, CP-038
	CP-010, CP-				
	011, CP-012,				
	CP-013, CP-				
	014, CP-039,				
	CP-040, CP-				
	041, CP-042,				
	CP-043, CP-				
	044, CP-045,				
	CP-046, CP-				
	047				
TOTAL	23	6	7	4	9

---

### 9.1.2.3. Plan de entregas

La Tabla 30 muestra el plan de entregas, en ella se puede visualizar cuales historias de usuario se llevarán a cabo en cada iteración y cuantas semanas durarán para su implementación.

Tabla 30 Plan de entregas

Iteración	1			2			3			4			5		
Historias	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 5	Sem. 6	Sem. 7	Sem. 8	Sem. 9	Sem. 10	Sem. 11	Sem. 12	Sem. 13	Sem. 14	Sem. 15
H1	■	■													
H2	■	■													
H3			■												
H4				■	■										
H5													■		
H6							■								
H7							■								
H8														■	■
H9											■	■			
H10											■	■			
H11								■	■						
H12										■					
H13				■											
H14													■	■	■
H15															■
H16	■														
H17				■											

### 9.1.3. Fase de Iteraciones

La fase de iteraciones comprende la codificación de las cinco iteraciones definidas anteriormente en el plan de iteraciones que se encuentra en el inciso 8.1.1.3.

Cada historia de usuario se abordó en 4 ítems: Primero, la planeación de las tareas en tarjetas Kanban donde se definieron las actividades para llevar a cabo la historia de usuario. Segundo, definición del modelo de datos; donde se definieron las entidades que participan, sus atributos y respectivas relaciones. Tercero, diseño de mockups; donde se establecieron las interfaces de usuario y cuarto la definición y validación de casos de pruebas.

La arquitectura de desarrollo inicial establecida es la arquitectura hexagonal (Ver Anexo 7). Esta arquitectura se encuentra compuesta en 3 capas: Infraestructura, aplicación y dominio. La capa de infraestructura es la parte que contiene todo lo que la aplicación necesita para funcionar, como la configuración de la base de datos o la configuración de Spring. La capa de aplicación, el usuario o cualquier otro programa interactúa con la aplicación. Esta área debe contener elementos como interfaces de usuario, controladores REST y bibliotecas de serialización JSON. La capa de dominio es el núcleo de nuestra aplicación, contiene toda la lógica del negocio. (*Arquitectura Hexagonal, DDD y Spring / Baeldung, n.d.*)

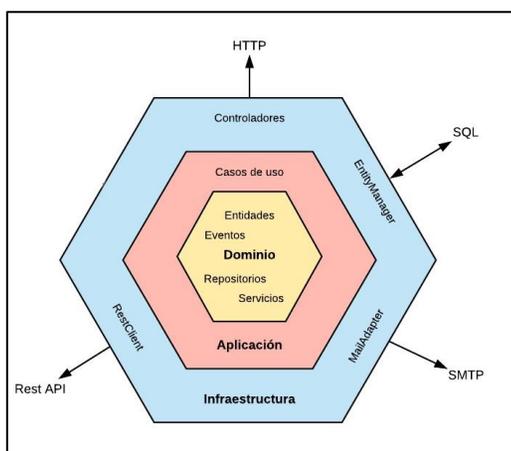


Figura 4 Arquitectura de desarrollo hexagonal

### 9.1.3.2. Iteración 1

La iteración 1 abarca las siguientes historias de usuario planteadas en la Tabla 31: gestionar las áreas académicas, gestionar usuarios, gestionar eventos y cursos de educación continua y realizar inscripción.

Tabla 31 Historias de usuario Iteración 1

Historia de Usuario	Desarrollo
HU16 gestionar las áreas académicas	Ver Anexo 1-1
HU1 gestionar usuarios	Ver Anexo 1-2
HU2 gestionar eventos y cursos de educación	Ver Anexo 1-3
continua	
HU3 realizar inscripción	Ver Anexo 1-4

### 9.1.3.3. Iteración 2

La iteración 2 abarca las siguientes historias de usuario planteadas en la Tabla 32: Generar reportes de SNIES, Cancelar Inscripción y Login Sistema

Tabla 32 Historias de usuario Iteración 2

Historia de Usuario	Desarrollo
HU4 Generar reportes de SNIES	Ver Anexo 2-1
HU13 Cancelar Inscripción	Ver Anexo 2-2
HU17 Login Sistema	Ver Anexo 2-3

Es importante resaltar que el login de los diferentes usuarios de sistema se realiza mediante la API de Google, por lo tanto se realizó la implementación de dicha API durante el desarrollo de esta iteración (Ver Anexo 2-4)

#### 9.1.3.4. Iteración 3

La iteración 3 abarca las siguientes historias de usuario planteadas en la Tabla 33: Gestionar ponentes de la educación continua, Gestionar jornadas de la educación continua y Generar listado de asistentes.

Tabla 33 Historias de usuario Iteración 3

Historia de Usuario	Desarrollo
HU6 Gestionar ponentes de la educación continua	Ver Anexo 3-1
HU7 Gestionar jornadas de la educación continua	Ver Anexo 3-2
HU11 Generar listado de asistentes	Ver Anexo 3-3

#### 9.1.3.5. Iteración 4

La iteración 4 abarca las siguientes historias de usuario planteadas en la Tabla 34: Generar certificados de participación, Obtener certificados de participación y Gestionar asistencia de los asistentes inscritos.

Tabla 34 Historias de usuario Iteración 4

Historia de Usuario	Desarrollo
HU9 Generar certificados de participación	Ver Anexo 4-1
HU10 Obtener certificados de participación	Ver Anexo 4-2
HU12 Gestionar asistencia de los asistentes inscritos	Ver Anexo 4-3

### 9.1.3.6. Iteración 5

La iteración 5 abarca las siguientes historias de usuario planteadas en la Tabla 35: Gestionar roles de usuarios, Visualizar datos estadísticos, Toma de asistencias mediante QR y Notificación vía email.

Tabla 35 Historias de usuario Iteración 5

Historia de Usuario	Desarrollo
HU5 Gestionar roles de usuarios	Ver Anexo 5-1
HU8 Visualizar datos estadísticos	Ver Anexo 5-2
HU14 Toma de asistencias mediante QR	Ver Capitulo 11
HU15 Notificación vía email	Ver Anexo 5-3

### 9.1.4. Fase de Puesta en Producción

El despliegue del sistema se realizó en dos arquitecturas debido a la fase de desarrollo y a la fase de producción del sistema. Durante la fase de desarrollo se realizó el despliegue del sistema al finalizar las iteraciones mediante el servicio EC2 de *Amazon Web Services* (AWS), donde por medio de Docker se realizó la configuración de varios *containers* para establecer la siguiente arquitectura presentada en la Figura 5.

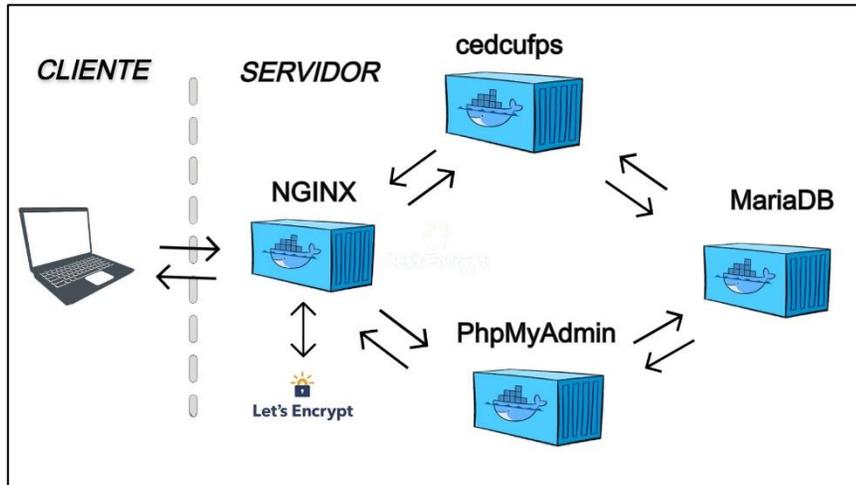


Figura 5. Arquitectura despliegue sistema fase desarrollo

Los contenedores implementados son descritos a continuación:

- Nginx Proxy

Es el contenedor que ejecuta nginx y docker-gen, esto genera configuraciones de proxy inverso para nginx para acceder mediante un dominio al sistema gestión de educación continua y a phpmyadmin.

- Let's Encrypt

Es un contenedor complementario ligero para nginx-proxy que maneja la creación, renovación y uso automatizados de certificados para el proxy.

- MariaDB

Es el contenedor que contiene el motor de base de datos utilizado para la persistencia de todos los datos gestionados en el sistema.

- PhpMyAdmin

Es el contenedor que contiene el sistema gestión de base datos utilizado para manejar la administración de MariaDB

- Cedcufps

Es el contenedor que contiene el sistema gestión de educación continua desarrollado. Para la creación de este contenedor se creó el respectivo Dockerfile (Ver Anexo 8)

Una vez definida la arquitectura de despliegue en fase de desarrollo, se realizó el proceso de instalación de los contenedores en el servidor (Ver Anexo 9)

Durante la fase de producción para el despliegue final del aplicativo se estableció la siguiente arquitectura en capas como se presenta en la Figura 6, utilizando los servicios de nginx proxy inverso, motor de base de datos MariaDB y el sistema gestión para la educación continua.

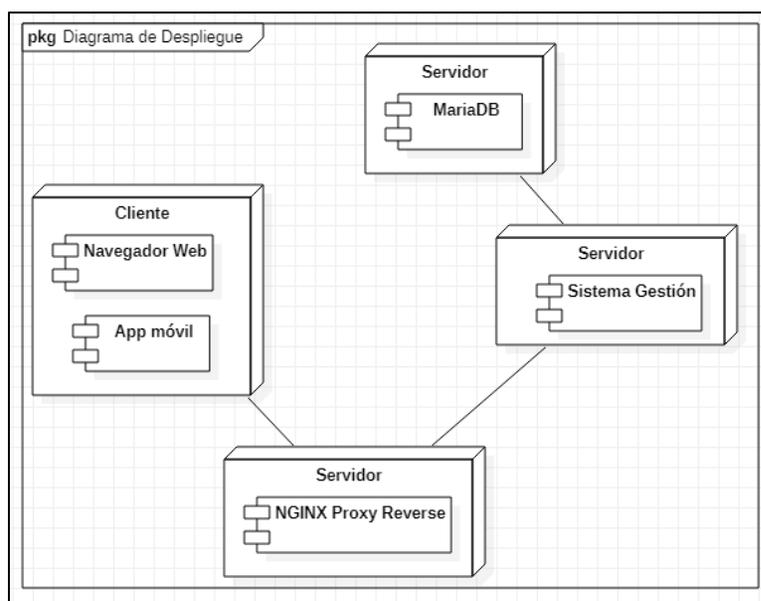


Figura 6. Arquitectura despliegue sistema fase de producción

## 10. Estrategia de creación de certificados de participación

Para la creación de certificados de participación se realizó una investigación entre las librerías javascript que utilizaban canvas y se optó por la implementación de la librería Fabric JS ya que proporciona un modelo de objeto interactivo sobre el elemento de lienzo permitiéndole al usuario final diseñar sus propios certificados sin necesidad de tener una estructura cerrada; es decir, limitada a cierta cantidad de objetos. Además, se tiene la posibilidad de arrastrarlos y redimensionarlos a través del lienzo. Para cumplir con el objetivo del diseño de certificados, los objetos utilizados fueron: Textbox para el manejo de textos, Image para el manejo de imágenes incluido la de fondo, y Line para el manejo de líneas para incluir las firmas.

Se definió una estructura base para plantear un diseño inicial que puede ser modificado posteriormente por el docente responsable de la educación continua. Para este diseño se establecieron unas etiquetas en la Tabla 36 con el fin de reemplazar algunos datos que tiene el evento académico.

Tabla 36 Etiquetas plantilla certificados de participación

Etiqueta	Dato
%facultad%	Nombre de la facultad en la cual se lleva a cabo la educación continua

%programa%	Nombre del programa académico en la cual se lleva a cabo la educación continua
%nombreAsistente%	Nombre completo del asistente a la educación continua
%documentoAsistente%	Tipo de documento y número de documento del asistente a la educación continua
%tipoEducacionContinua%	Tipo de la educación continua
%intensidadHoraria%	Intensidad horaria de la educación continua en texto
%fecha%	Fecha en la que se llevó a cabo la educación continua

---

Adicionalmente a estas etiquetas y como se mencionó anteriormente, el docente responsable de la educación continua podrá agregar textos, imágenes y líneas como se presenta a continuación:

- En la figura 7 se presenta la manera como se agregan imágenes a la plantilla del certificado de participación, además se puede observar que se puede elegir la imagen para colocarla de fondo de plantilla.

Figura 7 Interfaz agregar imágenes a certificados de participación

- En la figura 8 se presenta la manera como se agregan textos a la plantilla del certificado de participación, además se puede elegir los estilos del texto que se deseen como: la fuente, el tamaño, color, color de borde, negrita, cursiva, subrayado y/o tachado.

Figura 8 Interface agregar textos a certificados de participación

- En la figura 9 se presenta la manera como se agregan líneas a la plantilla del certificado de participación, además se puede elegir un color si se desea.

Figura 9 Interface agregar líneas a certificados de participación

En la figura 10 se presenta un ejemplo del resultado de la plantilla de certificado de participación de la educación continua dentro del sistema.



Figura 10 Resultado plantilla certificado de participación

## 11. Desarrollo aplicación móvil toma de asistencias

Para el desarrollo de la app móvil se definió tomar la metodología XP aplicada anteriormente.

### 11.1. Fase de exploración

Se definen en la tabla 37 las historias de usuario para la construcción de la aplicación para sistemas operativos Android.:

Tabla 37 Historias de usuario aplicación móvil

ID Historia de Usuario	Nombre
HU-A1	Login de la app
HU-A2	Listar educaciones continuas
HU-A3	Tomar asistencia mediante QR

A continuación, se detallarán cada historia de usuario con sus criterios de aceptación y los casos de prueba que se llevarán a cabo.

#### **HU-A1: Login de la app**

Como usuario con rol super-admin, manaeccu o attendance, quiero loguearme en la aplicación móvil mediante el email ya registrado.

#### Criterios de aceptación:

\*El usuario que desea ingresar debe tener alguno de los roles super-admin, manaeccu o attendance.

\*El usuario que no cuente con ninguna de los permisos admitidos, no podrá ingresar a la aplicación y deberá recibir un mensaje de ingreso fallido.

Definición casos de pruebas:

A continuación, se definirán en la Tabla 38 los diferentes casos de pruebas de la HU-A1, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 38 Casos de prueba HU-A1

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-A001	Login exitoso a la aplicación	El usuario con rol super-admin, manaeccu o attendance se loguea a la aplicación exitosamente
CP-A002	Login fallido a la aplicación	El usuario con un rol distinto a super-admin, manaeccu o attendance no puede loguearse a la aplicación

**HU-A2: Listar educaciones continuas**

Como usuario con rol super-admin, manaeccu o attendance, quiero visualizar las educaciones continuas que se encuentran en desarrollo.

Criterios de aceptación:

- \*El usuario una vez logeado, visualiza el listado de las educaciones continuas que se encuentran en desarrollo.
- \*El usuario no podrá visualizar educaciones continuas a las cuales no tiene permiso acceder.
- \*El usuario podrá seleccionar la educación continua y elegir la jornada a la cual se va a tomar la asistencia de los participantes

Definición casos de pruebas:

A continuación, se definirán en la Tabla 39 los diferentes casos de pruebas de la HU-A2, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 39 Casos de prueba HU-A2

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-A003	Visualizar listado de educaciones continua	El usuario con rol super-admin, manaeccu o attendance se logea a la aplicación exitosamente
CP-A004	Visualizar jornadas de la educación continua	El usuario con un rol distinto a super-admin, manaeccu o attendance no puede loguearse a la aplicación

**HU-A3: Tomar asistencia mediante QR**

Como usuario con rol super-admin, manaeccu o attendance, quiero tomar la asistencia de los participantes mediante su código QR generado al realizar la inscripción.

*Criterios de aceptación:*

- \*El participante debe mostrar su tarjeta de participación para que se código QR sea capturado.
- \*El usuario con alguno de los roles super-admin, director de programa o docente debe capturar el código QR de los participantes y mostrar un mensaje si se realizó exitosamente el proceso.
- \*Si un código QR es inválido o antiguo, deberá la aplicación mostrar un mensaje de error al momento de capturar el código QR

*Definición casos de pruebas:*

A continuación, se definirán en la Tabla 40 los diferentes casos de pruebas de la HU-A3, que se llevarán a cabo en la fase de desarrollo y que están acordes a los criterios de aceptación definidos anteriormente.

Tabla 40 Casos de prueba HU-A3

ID	NOMBRE CASO PRUEBA	DESCRIPCIÓN
CP-A005	Toma de asistencias a través de la app móvil	El usuario toma la lectura del código QR del participante para registrar la asistencia

CP-A006	Toma de asistencia de un código QR inválido a través de la app móvil	El usuario toma la lectura del código QR que es antiguo o no corresponde a la inscripción del participante
---------	--	--

---

## 11.2. Fase de Planificación

La tabla 41 presenta la estimación que se realizó de las historias de usuario de acuerdo a la metodología XP.

Tabla 41 Estimación historias de usuario aplicación móvil

Historia de Usuario	Nombre	Prioridad	Esfuerzo
HU-A1	Login de la app	Media	2
HU-A2	Listar educaciones continuas	Baja	1
HU-A3	Tomar asistencia mediante QR	Alta	2

En la Tabla 42 se encuentra el plan de iteraciones para el desarrollo de la aplicación móvil, donde se indica que historias de usuario se realizarán por cada una de las iteraciones.

Tabla 42 Plan de iteraciones aplicación móvil

Iteración	Historias de Usuario
Iteración 1	HU-A1, HU-A2, HU-A3

En la Tabla 43 se presenta los casos de pruebas que se mencionaron anteriormente y que se realizarán por cada una de las iteraciones de desarrollo.

Tabla 43 Casos de prueba por iteración aplicación móvil

ITERACIÓN 1	
ID PRUEBA	CP-A001, CP-A002, CP-A003, CP-A004, CP-A005, CP-A006
TOTAL	6

### 11.3. Fase de Iteraciones

Cada historia de usuario se abordó en 4 ítems: Primero, la planeación de las tareas en tarjetas Kanban donde se definieron las actividades para llevar a cabo la historia de usuario. Segundo, definición del modelo de datos; donde se definieron las entidades que participan, sus atributos y respectivas relaciones. Tercero, diseño de mockups; donde se establecieron las interfaces de usuario y cuarto la definición y validación de casos de pruebas unitarias y de aceptación.

#### 11.3.1. Iteración 1

La iteración 1 abarca las siguientes historias de usuario planteadas en la Tabla 35: Login de la app, listar educaciones continuas y tomar asistencia mediante QR. Se implementó la API de Google para el login de la aplicación mediante el email (Ver Anexo 10-1)

Tabla 44 Historias de usuario Iteración 1 aplicación móvil

Historia de Usuario	Desarrollo
HU-A1 Login de la app	Ver Anexo 10-2
HU-A2 Listar educaciones continuas	Ver Anexo 10-3

HU-A3 Tomar asistencia mediante QR      Ver Anexo 10-4

---

## 12. Migración de información

### ETAPA 1. ANALIZAR:

En esta etapa se evaluará y comprenderán los datos ya existentes para así enviarlos al nuevo sistema. Se determinará su calidad y origen para detectar los posibles errores.

Por esto se realizó el análisis del modelo de datos de la aplicación ADEEX en el cual al momento de realizar el estudio de este se evidenciaron:

- Tabla User: No se registran género, fecha de nacimiento, fecha de expedición de documento y estado civil, no se tiene segundo nombre ya que este registra de manera única los dos nombres en un solo campo y el correo electrónico debe ser único para cada usuario registrado.

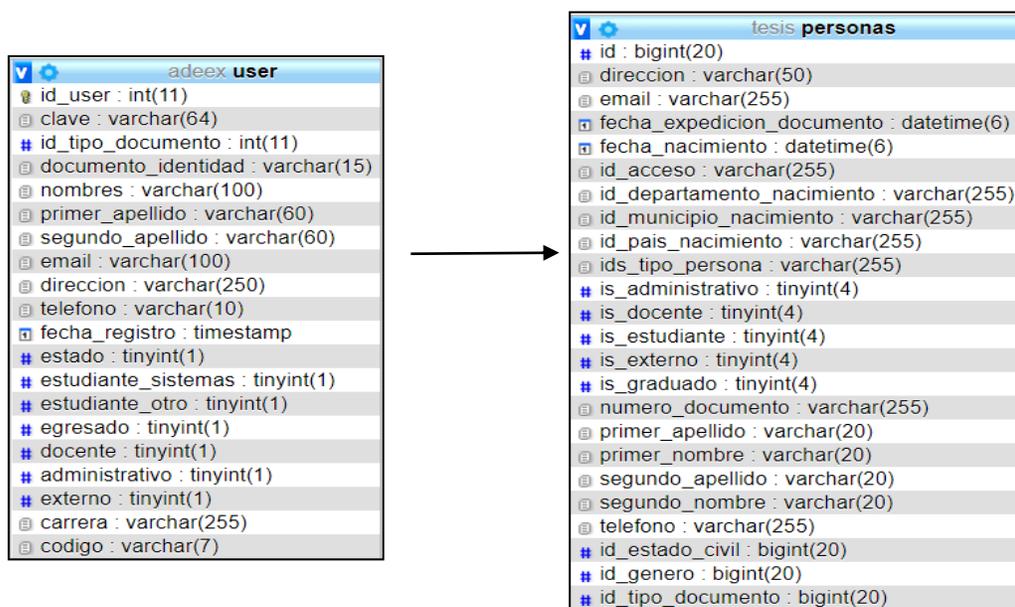


Figura 11 Comparación entidad Personas entre ADEEX y GEDUCO

- Tabla Evento: No se tiene el consecutivo el cual se crea con cada tipo de educación continua y en su respectivo programa.

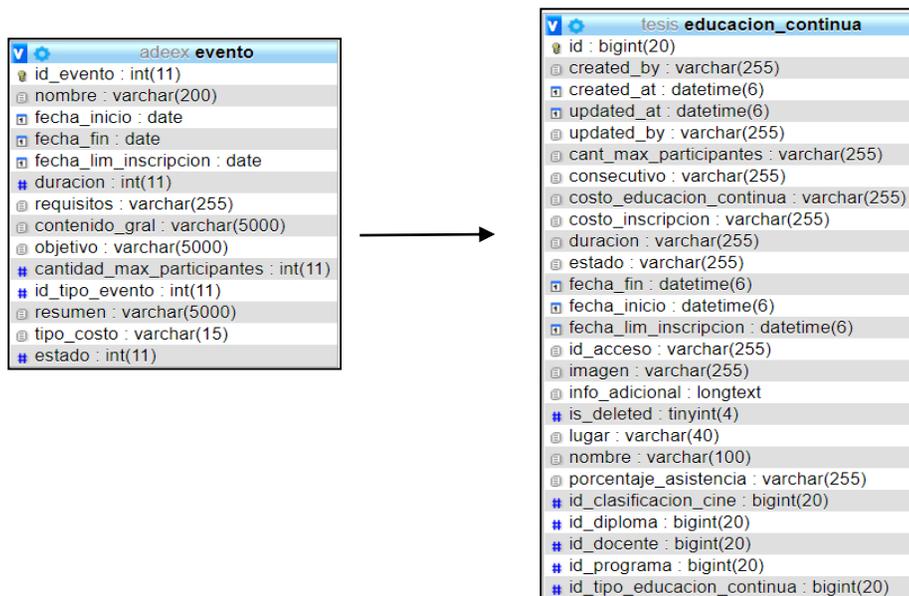


Figura 12 Comparación entidad Educación Continua entre ADEEX y GEDUCO

- Tabla Inscripción: Falta el campo token que es respectivo de cada participante.

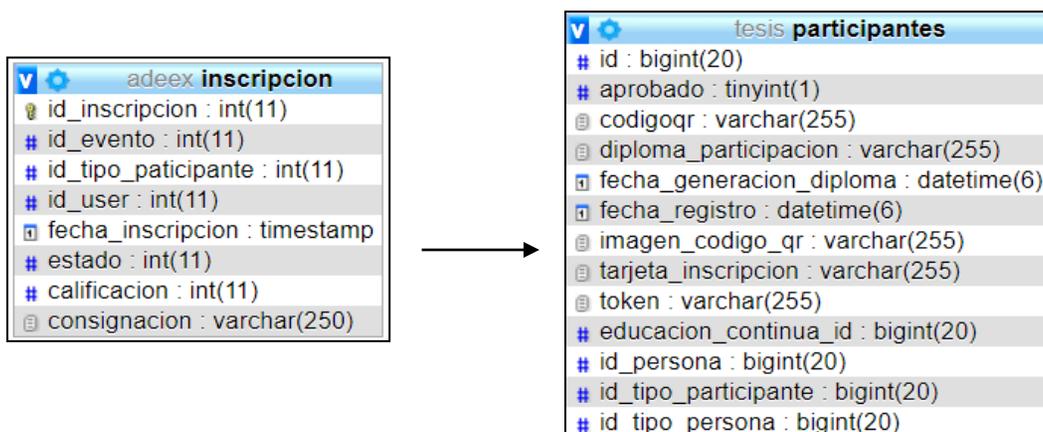


Figura 13 Comparación entidad Participantes entre ADEEX y GEDUCO

- Tabla Horario: Se tiene toda la información necesaria para hacer la migración

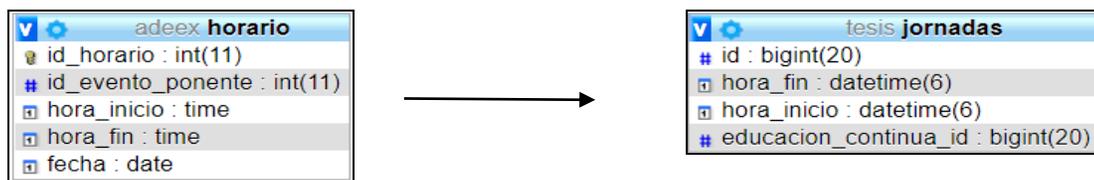


Figura 14 Comparación entidad Jornadas entre ADEEX y GEDUCO

- Tabla Control de Asistencias: Se tiene toda la información necesaria para hacer la migración

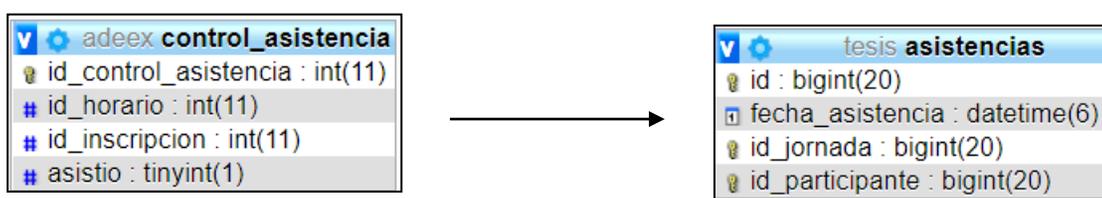


Figura 15 Comparación entidad Asistencias entre ADEEX y GEDUCO

- Tabla Tipo Documento: Se tiene diferente Id a los tipos de documento existentes.

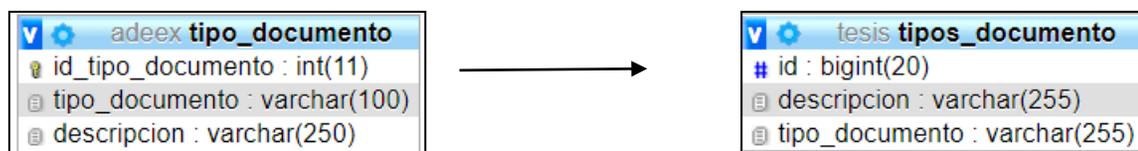


Figura 16 Comparación entidad Tipos Documento entre ADEEX y GEDUCO

- Tabla Tipo Evento: Se tienen más tipos de educación continua.

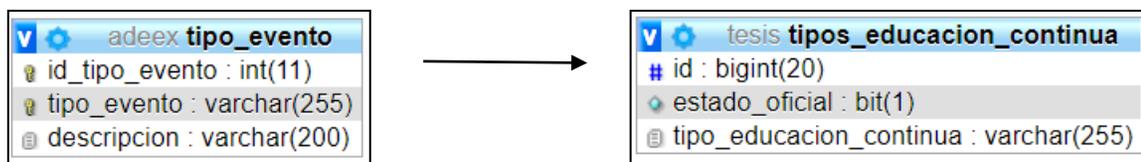


Figura 17 Comparación entidad Tipos Educación Continua entre ADEEX y GEDUCO

## ETAPA 2. EXTRAER / TRANSFORMAR

En esta etapa se transformarán los datos para adecuarlos de tal modo que sea necesario para el sistema nuevo. Las transformaciones que se tienen que realizar para así poder realizar una correcta migración serían:

- Tabla User: Se agregan los campos género, fecha de nacimiento, fecha de expedición de documento y estado civil de manera nula. Se dividen en 2 los nombres para así poderlos migrar de una manera más exacta.
- Tabla Evento: Se crea el campo consecutivo y se le da un valor único a cada evento registrado.
- Tabla Inscripción: Se añade el campo de token y se le da un valor único a cada inscripción realizada.

Para finalizar esta etapa se crean los scripts sql para ejecutar la correcta migración con las transformaciones ya realizadas. (Ver Anexo 11)

### ETAPA 3. CARGAR / VALIDAR

En esta etapa ya estarán los datos convertidos para ser cargados al nuevo sistema. Se realizará de manera directa para así poder tener una validación. Si ocurre algún fallo se deberá volver a las anteriores etapas para revisar las causas de este fallo. Para desarrollar esta etapa se ejecutan los scripts creados en la etapa anterior en la base de datos del proyecto final. Después de ejecutar los scripts validamos que todos los datos estén en el aplicativo de manera correcta. A continuación, se verán todas las educaciones continuas realizadas hasta la fecha mediante el software ADEEX.

EDUCACIONES CONTINUAS <span style="float: right;">+</span>						
El siguiente listado son las educaciones continuas a los cuales usted tiene permisos de gestionar						
Mostrar <input type="text" value="10"/> registros		Buscar: <input type="text"/>				
Estado	Tipo	Programa Académico Organizador	Nombre	Fecha Inicio	Fecha Limite Inscripción	
	Conferencia	Ingeniería de Sistemas	CaféTic	03-05-2021 18:00	14-05-2021 12:00	
	Seminario	Ingeniería de Sistemas	X ENCUENTRO DE SEMILLEROS DE INVESTIGACION: Feria de Proyectos de Aula Virtual	10-06-2020 05:00	09-06-2020 05:00	
	Taller	Ingeniería de Sistemas	Como Investigar con Datos Abiertos? Generando conocimiento con datos abiertos	16-01-2020 05:00	14-01-2020 05:00	
	Curso	Ingeniería de Sistemas	Prototipado Web y Móvil	14-11-2019 05:00	13-11-2019 05:00	
	Seminario	Ingeniería de Sistemas	VIII Encuentro Semilleros y Proyectos Aula	27-11-2019 05:00	18-11-2019 05:00	
	Curso	Ingeniería de Sistemas	Frontend JavaScript y Php	19-11-2019 05:00	18-11-2019 05:00	
	Curso	Ingeniería de Sistemas	Backend Python	19-11-2019 05:00	18-11-2019 05:00	
	Conferencia	Ingeniería de Sistemas	CAFETIC -Programación asíncrona en springboot	17-10-2019 05:00	15-10-2019 05:00	
	Curso	Ingeniería de Sistemas	Análisis de datos con Phytón.	04-10-2019 05:00	03-10-2019 05:00	
	Curso	Ingeniería de Sistemas	Arquitectura de Microservicio con Spring Cloud Netflix y React JS como medio para consumir servicio REST	30-05-2019 05:00	29-05-2019 05:00	

Mostrando registros del 1 al 10 de un total de 55 registros

Anterior  2 3 4 5 6 Siguiente

Figura 18 Página 1 de educaciones continuas migradas

EDUCACIONES CONTINUAS +

El siguiente listado son las educaciones continuas a los cuales usted tiene permisos de gestionar

Mostrar 10 registros

Estado	Tipo	Programa Académico Organizador	Nombre	Fecha Inicio	Fecha Limite Inscripción
		Ingeniería de Sistemas	Introducción a la Ciencia de Datos	26-04-2019 05:00	25-04-2019 05:00
		Ingeniería de Sistemas	Introducción a Python para la Ciencia de Datos	26-04-2019 05:00	25-04-2019 05:00
		Ingeniería de Sistemas	MANTENIMIENTO DE HARDWARE Y SOFTWARE	06-04-2019 05:00	01-04-2019 05:00
		Ingeniería de Sistemas	Encuentro - Egresados	06-12-2018 05:00	05-12-2018 05:00
		Ingeniería de Sistemas	Integración Estudiantes	26-10-2018 05:00	24-10-2018 05:00
		Ingeniería de Sistemas	Curso Herramienta Informática: Excel de nivel intermedio-avanzado Jornada: Mañana	13-11-2018 05:00	09-11-2018 05:00
		Ingeniería de Sistemas	Curso Herramienta Informática: Excel de nivel intermedio-avanzado	13-11-2018 05:00	09-11-2018 05:00
		Ingeniería de Sistemas	4a. versión del Encuentro Internacional de Ciencias Aplicadas e Ingeniería (Universidad - Industria) EISI 2018	06-09-2018 05:00	05-09-2018 05:00
		Ingeniería de Sistemas	Feria de Proyectos de Aula 3.0	21-06-2018 05:00	20-06-2018 05:00
		Ingeniería de Sistemas	Feria de Proyectos de Aula 2.0	29-11-2017 05:00	25-11-2017 05:00

Mostrando registros del 11 al 20 de un total de 55 registros

Anterior 1 2 3 4 5 6 Siguiente

Figura 19 Página 2 de educaciones continuas migradas

EDUCACIONES CONTINUAS +

El siguiente listado son las educaciones continuas a los cuales usted tiene permisos de gestionar

Mostrar 10 registros

Estado	Tipo	Programa Académico Organizador	Nombre	Fecha Inicio	Fecha Limite Inscripción
		Ingeniería de Sistemas	I Jornada de actualización en seguridad informática	30-11-2017 05:00	29-11-2017 05:00
		Ingeniería de Sistemas	VII ENCUENTRO DE SEMILLEROS DE INVESTIGACION DE INGENIERIA DE SISTEMAS	12-06-2019 05:00	30-05-2019 05:00
		Ingeniería de Sistemas	Congreso Internacional en Innovación y Apropiación de las Tecnologías de la Información y las Comunicaciones – CIINATIC	13-09-2017 05:00	11-09-2017 05:00
		Ingeniería de Sistemas	Programa de Acceso a Información Estadística de investigaciones del DANE	06-06-2017 05:00	05-06-2017 05:00
		Ingeniería de Sistemas	Tips y uso de herramientas abiertas para ciencia abierta	24-05-2017 05:00	23-05-2017 05:00
		Ingeniería de Sistemas	Programa "En Tic Confío" Uso Responsable de Internet y las Nuevas Tecnologías.	24-05-2017 05:00	23-05-2017 05:00
		Ingeniería de Sistemas	CONVOCATORIA PARA PARTICIPAR EN EL V ENCUENTRO DE SEMILLEROS DE INVESTIGACION DE INGENIERIA DE SISTEMAS	08-06-2017 05:00	02-06-2017 05:00
		Ingeniería de Sistemas	B-learning - construcción de MOOC	05-05-2017 05:00	04-05-2017 05:00
		Ingeniería de Sistemas	B-learning, construcción de un MOOC.	05-05-2017 05:00	28-04-2017 05:00
		Ingeniería de Sistemas	PROGRAMACION BASICA EN JAVA	05-05-2017 05:00	04-05-2017 05:00

Mostrando registros del 21 al 30 de un total de 55 registros

Anterior 1 2 3 4 5 6 Siguiente

Figura 20 Página 3 de educaciones continuas migradas

EDUCACIONES CONTINUAS +

El siguiente listado son las educaciones continuas a los cuales usted tiene permisos de gestionar

Mostrar 10 registros  Buscar:

Estado	Tipo	Programa Académico Organizador	Nombre	Fecha Inicio	Fecha Limite Inscripción
	Curso	Ingeniería de Sistemas	MANTENIMIENTO DE HARDWARE Y SOFTWARE	01-07-2017 05:00	28-06-2017 05:00
	Curso	Ingeniería de Sistemas	AUTOCAD CON ENFASIS MINERO	23-01-2017 05:00	21-01-2017 05:00
	Seminario	Ingeniería de Sistemas	IV ENCUENTRO DE SEMILLEROS DE INVESTIGACION DE INGENIERIA DE SISTEMAS	17-11-2016 05:00	14-11-2016 05:00
	Curso	Ingeniería de Sistemas	EDICION DE IMÁGENES PARA MARKETING DIGITAL	12-05-2017 05:00	09-05-2017 05:00
	Curso	Ingeniería de Sistemas	DISEÑO WEB PARA PRINCIPIANTES	28-04-2017 05:00	24-04-2017 05:00
	Curso	Ingeniería de Sistemas	MANTENIMIENTO DE HARDWARE Y SOFTWARE	12-11-2016 05:00	11-11-2016 05:00
	Seminario	Ingeniería de Sistemas	III Encuentro de Semilleros de Investigación	25-05-2016 05:00	24-05-2016 05:00
	Seminario	Ingeniería de Sistemas	III Encuentro Semilleros de Investigación	03-12-2015 05:00	02-12-2015 05:00
	Autoevaluacion	Ingeniería de Sistemas	Encuentro - Estudiantes	26-11-2015 05:00	20-11-2015 05:00
	Autoevaluacion	Ingeniería de Sistemas	Encuentro - Egresados	04-12-2015 05:00	02-12-2015 05:00

Mostrando registros del 31 al 40 de un total de 55 registros

Anterior 1 2 3 4 5 6 Siguiente

Figura 22 Página 4 de educaciones continuas migradas

EDUCACIONES CONTINUAS +

El siguiente listado son las educaciones continuas a los cuales usted tiene permisos de gestionar

Mostrar 10 registros  Buscar:

Estado	Tipo	Programa Académico Organizador	Nombre	Fecha Inicio	Fecha Limite Inscripción
	Curso	Ingeniería de Sistemas	FUNDAMENTOS Y SEGURIDAD INFORMATICA CON LINUX	05-05-2017 05:00	02-05-2017 05:00
	Curso	Ingeniería de Sistemas	GERENCIA DE PROYECTOS	05-05-2017 05:00	24-04-2017 05:00
	Curso	Ingeniería de Sistemas	ANALITICA Y MINERIA DE DATOS DEL NEGOCIO	05-05-2017 05:00	28-04-2017 05:00
	Curso	Ingeniería de Sistemas	INTELIGENCIA DE NEGOCIOS	11-11-2016 05:00	10-11-2016 05:00
	Taller	Ingeniería de Sistemas	Plataforma SUNN	06-11-2015 05:00	05-11-2015 05:00
	Curso	Ingeniería de Sistemas	Introducción a la Ingeniería de requisitos	15-05-2015 05:00	14-05-2015 05:00
	Curso	Ingeniería de Sistemas	DISEÑO DE INTERFACES DE USUARIO (UX)	05-05-2017 05:00	03-05-2017 05:00
	Curso	Ingeniería de Sistemas	Inteligencia de Negocios (Business Intelligence)	05-06-2015 05:00	04-06-2015 05:00
	Curso	Ingeniería de Sistemas	AUTOCAD 2D	10-04-2015 05:00	09-04-2015 05:00
	Curso	Ingeniería de Sistemas	DESARROLLO DE APLICACIONES UTILIZANDO PHP	18-04-2017 05:00	17-04-2017 05:00

Mostrando registros del 41 al 50 de un total de 55 registros

Anterior 1 2 3 4 5 6 Siguiente

Figura 21 Página 5 de educaciones continuas migradas

EDUCACIONES CONTINUAS +

El siguiente listado son las educaciones continuas a los cuales usted tiene permisos de gestionar

Mostrar 10 registros Buscar:

Estado	Tipo	Programa Académico Organizador	Nombre	Fecha Inicio	Fecha Limite Inscripción
	Curso	Ingeniería de Sistemas	CONSTRUCCION DE OBJETOS VIRTUALES DE APRENDIZAJE (OVA)	04-05-2017 05:00	03-05-2017 05:00
	Curso	Ingeniería de Sistemas	MANTENIMIENTO PREVENTIVO Y CORRECTIVO DE HARDWARE Y SOFTWARE DE COMPUTADORES	16-03-2016 05:00	15-03-2016 05:00
	Curso	Ingeniería de Sistemas	Latex Essentials	12-06-2015 05:00	10-06-2015 05:00
	Curso	Ingeniería de Sistemas	Latex Essentials	14-11-2014 05:00	13-11-2014 05:00
	Curso	Ingeniería de Sistemas	MANTENIMIENTO PREVENTIVO Y CORRECTIVO DE HARDWARE Y SOFTWARE DE COMPUTADORES	07-11-2014 05:00	06-11-2014 05:00

Mostrando registros del 51 al 55 de un total de 55 registros Anterior 1 2 3 4 5 6 Siguiente

Figura 23 Página 6 de educaciones continuas migradas

A continuación, se visualizarán todas las educaciones continuas realizadas por los usuarios registrados junto con sus diplomas.

Inicio Eventos Inscrito **Certificados** Javier Calderon

MIS CERTIFICACIONES

A continuación se listarán las educaciones continuas a las cuales se ha inscrito. El certificado de participación se podrá descargar una vez haya sido aprobada su participación.

Mostrar 10 registros Buscar:

Certificado	Tipo	Educación Continua	Fecha Inicio	Tipo Participante
	Seminario	VII ENCUENTRO DE SEMILLEROS DE INVESTIGACION DE INGENIERIA DE SISTEMAS	12-06-2019 05:00	Asistente
	Curso	Arquitectura de Microservicio con Spring Cloud Netflix y React JS como medio para consumir servicio REST	30-05-2019 05:00	Asistente
	Congreso	4a. versión del Encuentro Internacional de Ciencias Aplicadas e Ingeniería (Universidad - Industria) EISI 2018	06-09-2018 05:00	Asistente
	Congreso	Congreso Internacional en Innovación y Apropiación de las Tecnologías de la Información y las Comunicaciones – CIINATIC	13-09-2017 05:00	Asistente
	Curso	DISEÑO DE INTERFACES DE USUARIO (UX)	05-05-2017 05:00	Asistente
	Curso	MANTENIMIENTO DE HARDWARE Y SOFTWARE	12-11-2016 05:00	Asistente

Mostrando registros del 1 al 6 de un total de 6 registros Anterior 1 Siguiente

Figura 24 Certificaciones de los cursos realizados por un usuario



Figura 25 Diploma de un curso realizado por un usuario

## Conclusiones

En la presente tesis de grado que lleva por título implementación de un sistema de información para la gestión de los cursos y actividades para educación continua de los programas académicos de la universidad Francisco de Paula Santander de la ciudad de Cúcuta, se pretende generar una herramienta que sirva de apoyo para dar cumplimiento a las políticas y metas de calidad del MEN, logrando implementar un sistema gestión de actividades de educación continua (cursos, talleres, diplomados, seminarios, congresos y simposios) que se llevan a cabo en los diferentes Programas Académicos de la Universidad. Teniendo en cuenta lo anterior se llegó a las siguientes conclusiones:

- El desarrollo de un método para la toma de asistencia utilizando el dispositivo móvil y código QR permite agilizar estos procesos.
- Se implementó un motor de certificados a través de la librería Fabric JS que permiten a los usuarios del sistema tener mucha flexibilidad al momento de generarlos.
- Los informes generados son basados en las directrices dadas por SNIES en su documento “La Estructura conceptual y funcional”, por lo tanto, este requerimiento es cerrado
- La automatización de la integración continua continua entre las herramientas DockerHub y GitHub permitió optimizar el proceso de despliegue en el servidor cloud del programa de Ingeniería de Sistemas

## Recomendaciones

Después de la implementación de una plataforma que permitiera agilizar los procesos de información y entrega de informes y certificados requeridos para cada uno de los eventos programados de educación continua y teniendo en cuenta los lineamientos del SNIES se ha considerado hacer las siguientes recomendaciones:

- Socializar a las docentes de la universidad en el uso de la plataforma GEDUCO.
- Promover la planificación cursos y educación continua en la UFPS a través de la plataforma GEDUCO.
- Evaluar permanentemente la funcionalidad de dicha plataforma en la búsqueda de una mejora continua, permitiendo que surjan nuevas investigaciones que contribuyan a ampliar la funcionalidad de esta.

### Referencias bibliográficas

- Baldeón, A., Steven, H., & Cazco, A. S. (2019). *SISTEMA INFORMÁTICO PARA AUTOMATIZAR EL CONTROL DE ASISTENCIA A CLASES DE DOCENTES Y ESTUDIANTES DE LA ESPOCH CON DISPOSITIVOS RFID, APLICANDO LA METODOLOGÍA SCRUMBAN*.
- Baridura, K. M., & Nwiabu, N. (2018). Developing an Analytic Staff Attendance Management System Using QR Code. *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND MATHEMATICAL THEORY*, 4(2), 1–11.
- Deng, M., & Feng, P. (2019). Research Status on RFID System Authentication Protocol. In *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science - AICS 2019* (pp. 432–438). New York, New York, USA: ACM Press.  
<https://doi.org/10.1145/3349341.3349446>
- El Mrabet, H., & Moussa, A. A. (2017). Research and design of smart management system in classroom. In *ACM International Conference Proceeding Series*. Association for Computing Machinery. <https://doi.org/10.1145/3175628.3175645>
- Escobar Chena, V. D. (2018). *Aplicación web para la gestión de boletos y ticketera móvil para cines*. Noordhoff.
- Gelvez Quintero, M. F. (2014). Implementación de un Sistema de Administración para los eventos de extensión del Departamento de Sistemas y el Programa de Ingeniería de Sistemas de la UFPS.
- Ministerio de Educación Nacional, C. (2017). Estructura Conceptual y Funcional - Manual de

Usuario, 1–277.

Ñustez Guzman, J. N. (2018). *Sistema para la integración del proceso de autenticación en aplicaciones web mediante códigos QR*.

Robles Ordóñez, S. E. (2018). *Software para la gestión del Programa de Innovación y Formación Docente de la UTPL*.

Sepúlveda, L., Yeiny, H., & Almentero, C. (2018). *DESARROLLAR UNA PLATAFORMA WEB PARA EL CONTROL DE ASISTENCIA DE ESTUDIANTES, DOCENTES Y ADMINISTRATIVOS DE LA UNIVERSIDAD DE CÓRDOBA SEDE LORICA MEDIANTE EL USO DE TARJETAS INTELIGENTES, DISPOSITIVOS MÓVILES Y SMS*.

Song, C., Li, Z., Xu, W., Zhou, C., Jin, Z., & Ren, K. (2018). My Smartphone Recognizes Genuine QR Codes! *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2), 1–20. <https://doi.org/10.1145/3214286>

Tiwari, S. (2016). An Introduction to QR Code Technology. In *2016 International Conference on Information Technology (ICIT)* (pp. 39–44). IEEE. <https://doi.org/10.1109/ICIT.2016.021>

Tucker, S., Darcy, P., & Stantic, B. (2014). A comparative study of RFID technology measuring efficiency and acceptance when capturing attendance. In B. Thomas & D. Parry (Eds.). Australian Computer Society, Inc.

Lee, S. D., Shin, M. K. y Kim, H. J. (2007). *EPC vs IPv6 mapping mechanism*. IEEE Conf. Publications, vol. 2, pp. 1243-1245

Miquel, S., F. Parra, C. Lhermie, and M. Miquel. (2006). *Distribución Comercial*.



*Usuario*. 1–277.

Montero Molina, B., Cevallos Vite, H., & Dávila Cuesta, J. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espiraes Revista Multidisciplinaria de Investigación ISSN: 2550-6862*, 2(17), 114–121.  
<https://doi.org/https://doi.org/10.1007/BF03194495>

Beck, Kent. 1999. *Extreme Programming Explained*. Addison-Wesley Professional

**ANEXOS**

## ANEXO 1. ITERACIÓN 1 FASE DE DESARROLLO

## ANEXO 1-1. HU16 Gestionar las áreas académicas

## Planeación de tareas en tarjeta Kanban

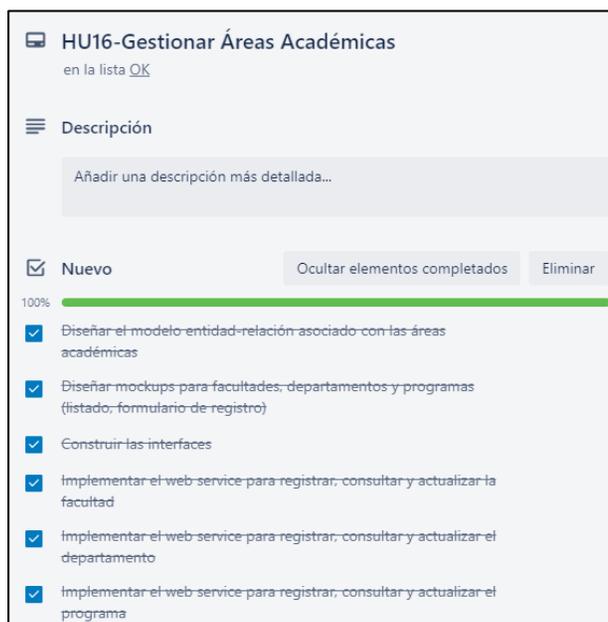


Figura 26 Tarjeta Kanban Gestionar áreas académicas.

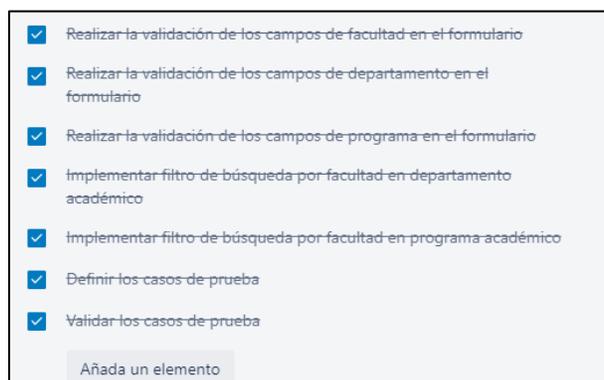


Figura 27 Tarjeta Kanban Gestionar áreas académicas.

## Definición del modelo de datos

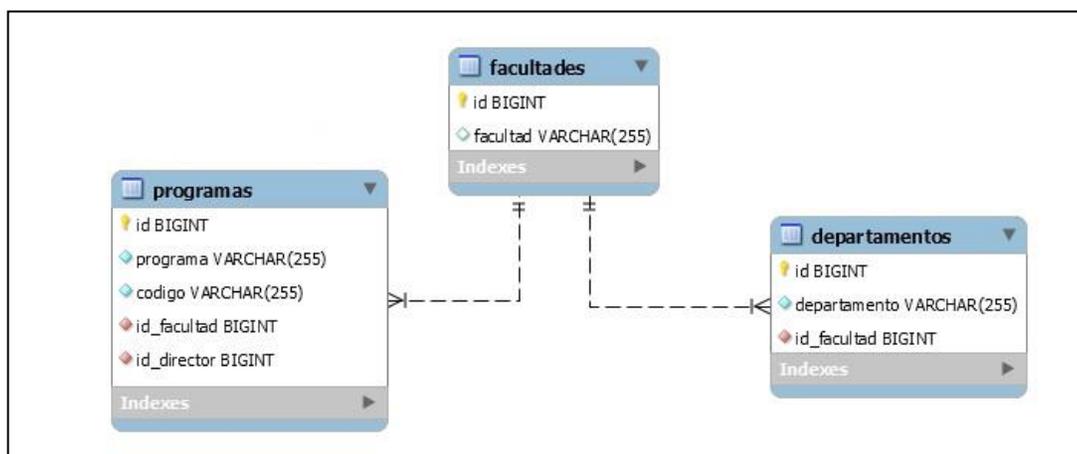


Figura 28 Modelo de datos áreas académicas

## Diseño de mockups

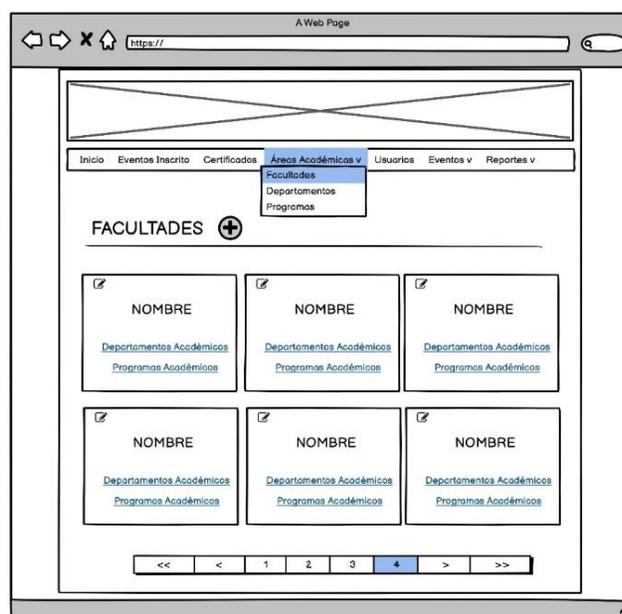


Figura 29 Mockup listar facultades

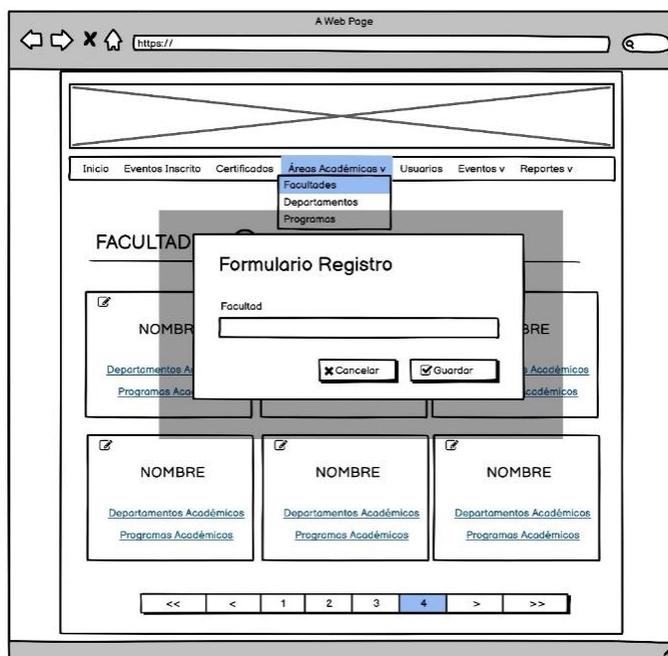


Figura 30 Mockup registrar facultad.

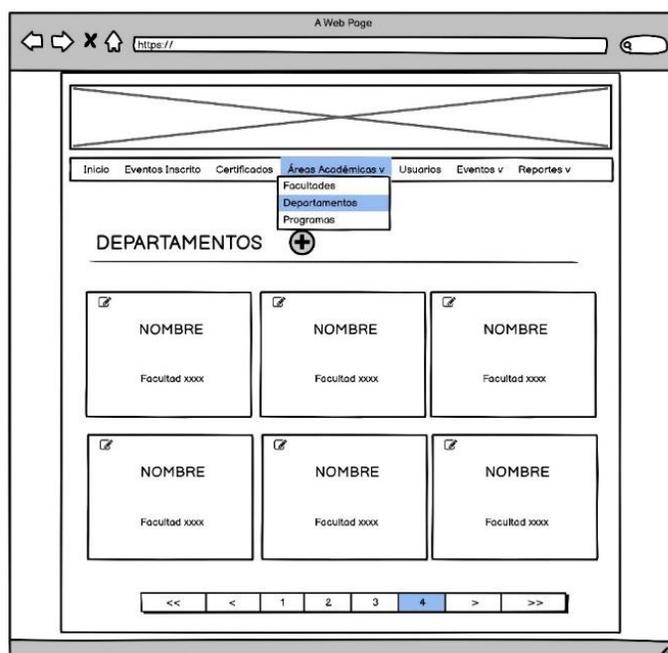


Figura 31 Mockup listar departamentos

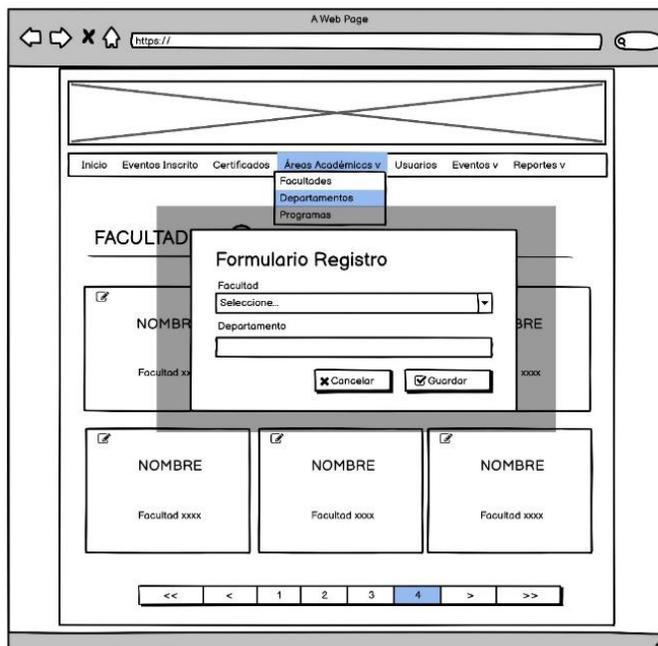


Figura 32 Mockup registro departamento

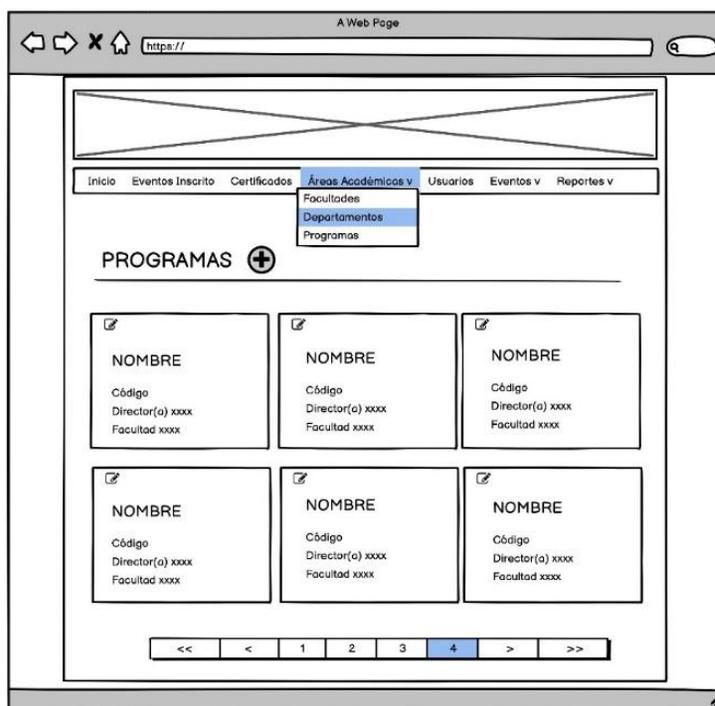


Figura 33 Mockup listar programas

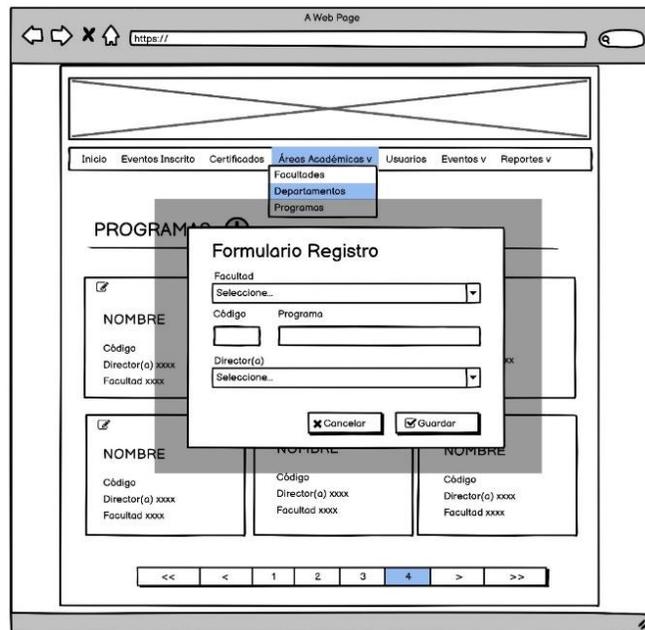


Figura 34 Mockup registrar programa.

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IFacultadService facultadService;

@InjectMocks
private FacultadRestController facultadController;

private static List<Facultad> LIST_FACULTADES;

private static Facultad FACULTAD1;
private static Facultad FACULTAD2;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    LIST_FACULTADES= new ArrayList<>();

    FACULTAD1=new Facultad(1L, "Ingeniería");
    FACULTAD2=new Facultad(2L, "Ciencias Empresariales");
    LIST_FACULTADES.add(FACULTAD1);
    LIST_FACULTADES.add(FACULTAD2);
}

```

Figura 35 Inicialización variables test gestionar facultades.

```

@Test
public void listFacultadTest() {
    Mockito.when(facultadService.findAll()).thenReturn(LIST_FACULTADES);
    final List<Facultad> response = facultadController.listar();
    assertNotNull(response);
    assertFalse(response.isEmpty());
    assertEquals(response.size(), 2);
}

@Test
public void findFacultadExistedTest() {
    Mockito.when(facultadService.findOne(1L)).thenReturn(FACULTAD1);
    final ResponseEntity<Facultad> response = facultadController.buscarPorFacultad(1L);
    assertNotNull(response.getBody());
}

@Test
public void findFacultadNotFoundTest() {
    Mockito.when(facultadService.findOne(1L)).thenReturn(FACULTAD1);
    final ResponseEntity<Facultad> response = facultadController.buscarPorFacultad(2L);
    assertNull(response.getBody());
}

@Test
public void saveFacultadTest() {
    final ResponseEntity<?> response = facultadController.guardarFacultadRest(FACULTAD1);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 38 Test gestionar facultades

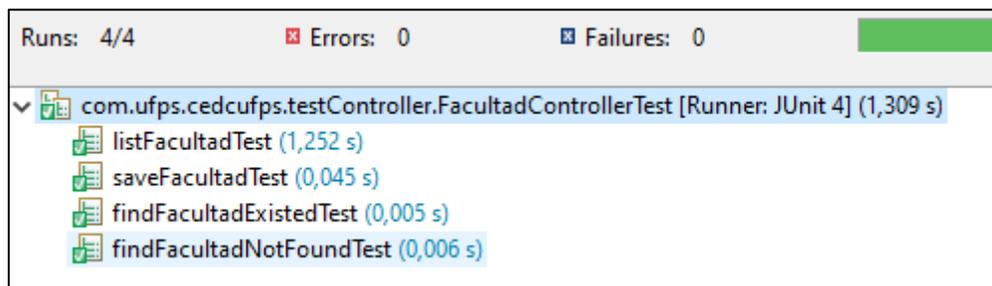


Figura 37 Resultados test gestionar facultades

```

@Mock
private IDepartamentoService departamentoService;

@InjectMocks
private DepartamentoRestController departamentoController;

private static List<Departamento> LIST_DEPARTAMENTOS;
private static Facultad FACULTAD1;
private static Departamento DEPARTAMENTO1;
private static Departamento DEPARTAMENTO2;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    LIST_DEPARTAMENTOS= new ArrayList<>();
    FACULTAD1=new Facultad(1L, "Ingeniería");
    DEPARTAMENTO1=new Departamento(1L, "Sistemas e Informática",FACULTAD1);
    DEPARTAMENTO2=new Departamento(2L, "Departamento Fluidos y Térmicas", FACULTAD1);
    LIST_DEPARTAMENTOS.add(DEPARTAMENTO1);
    LIST_DEPARTAMENTOS.add(DEPARTAMENTO2);
}

```

Figura 36 Inicialización variables test gestionar departamentos

```

@Test
public void listDepartamentoTest() {
    Mockito.when(departamentoService.findAll()).thenReturn(LIST_DEPARTAMENTOS);
    final List<Departamento> response = departamentoController.listar();
    assertNotNull(response);
    assertFalse(response.isEmpty());
    assertEquals(response.size(), 2);
}

@Test
public void findDepartamentoExistedTest() {
    Mockito.when(departamentoService.findOne(1L)).thenReturn(DEPARTAMENTO1);
    final ResponseEntity<Departamento> response = departamentoController.buscarPorFacultad(1L);
    assertNotNull(response.getBody());
}

@Test
public void findDepartamentoNotFoundTest() {
    Mockito.when(departamentoService.findOne(1L)).thenReturn(DEPARTAMENTO1);
    final ResponseEntity<Departamento> response = departamentoController.buscarPorFacultad(2L);
    assertNull(response.getBody());
}

@Test
public void saveDepartamentoTest() {
    final ResponseEntity<?> response = departamentoController.guardarDepartamentoRest(DEPARTAMENTO1);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 40. Test gestionar departamentos

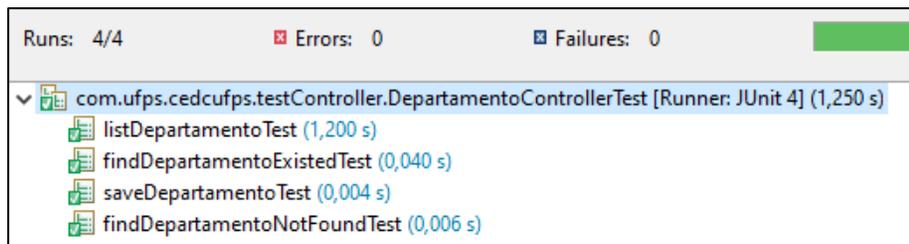


Figura 39. Resultados test gestionar departamentos

```

@Mock
private IProgramaService programaService;

@InjectMocks
private ProgramaRestController programaController;

private static List<Programa> LIST_PROGRAMAS;
private static Facultad FACULTAD1;
private static Persona DIRECTOR1;
private static Persona DIRECTOR2;
private static Programa PROGRAMA1;
private static Programa PROGRAMA2;

private static ProgramaDto PROGRAMA_DTO_1;
private static ProgramaDto PROGRAMA_DTO_2;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    LIST_PROGRAMAS = new ArrayList<>();
    FACULTAD1 = new Facultad(1L, "Ingeniería");
    DIRECTOR1 = new Docente();
    DIRECTOR1.setId(1L);
    DIRECTOR1.setPrimerNombre("Pilar");
    DIRECTOR1.setPrimerApellido("Rodriguez");

    DIRECTOR2 = new Docente();
    DIRECTOR2.setId(2L);
    DIRECTOR2.setPrimerNombre("Javier");
    DIRECTOR2.setPrimerApellido("Cardenas");
    PROGRAMA1 = new Programa(1L, "Ingeniería de Sistemas", FACULTAD1, (Docente)DIRECTOR1);
    PROGRAMA2 = new Programa(2L, "Ingeniería Civil", FACULTAD1, (Docente)DIRECTOR2);
    LIST_PROGRAMAS.add(PROGRAMA1);
    LIST_PROGRAMAS.add(PROGRAMA2);
}

```

Figura 41 Inicialización variables test gestionar programas

```

@Test
public void listProgramaTest() {
    Mockito.when(programaService.findAll()).thenReturn(LIST_PROGRAMAS);
    final List<Programa> response = programaController.listar();
    assertNotNull(response);
    assertFalse(response.isEmpty());
    assertEquals(response.size(), 2);
}

@Test
public void findProgramaExistedTest() {
    Mockito.when(programaService.searchProgramaById(1L)).thenReturn(PROGRAMA_DTO_1);
    final ResponseEntity<ProgramaDto> response = programaController.buscarPorPrograma(1L);
    assertNotNull(response.getBody());
}

@Test
public void findProgramaNotFoundTest() {
    Mockito.when(programaService.searchProgramaById(1L)).thenReturn(PROGRAMA_DTO_1);
    final ResponseEntity<ProgramaDto> response = programaController.buscarPorPrograma(2L);
    assertNull(response.getBody());
}

```

Figura 43. Test listar, buscar programas

```

@Test
public void saveProgramaTest() {
    final ResponseEntity<?> response = programaController.guardarProgramaRest(PROGRAMA1);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void buscarDirectorProgramaTest() {
    Mockito.when(programaService.findProgramaByDirector(1L,1L)).thenReturn(PROGRAMA_DTO_1);
    final ResponseEntity<ProgramaDto> response = programaController.buscarDirectorPrograma(
        PROGRAMA1.getDirectorPrograma().getId(),PROGRAMA1.getId());
    assertNotNull(response.getBody());
}

@Test
public void buscarDirectorProgramaNotFountTest() {
    Mockito.when(programaService.findProgramaByDirector(1L,1L)).thenReturn(PROGRAMA_DTO_1);
    final ResponseEntity<ProgramaDto> response = programaController.buscarDirectorPrograma(
        PROGRAMA2.getDirectorPrograma().getId(),PROGRAMA2.getId());
    assertNull(response.getBody());
}

```

Figura 42 Test guardar y buscar director de programa

```

Runs: 6/6      Errors: 0      Failures: 0
└─ com.ufps.cedcufps.testController.ProgramaControllerTest [Runner: JUnit 4] (1,525 s)
   └─ listProgramaTest (1,452 s)
   └─ saveProgramaTest (0,056 s)
   └─ findProgramaExistedTest (0,005 s)
   └─ buscarDirectorProgramaNotFountTest (0,004 s)
   └─ buscarDirectorProgramaTest (0,003 s)
   └─ findProgramaNotFoundTest (0,005 s)

```

Figura 44. Resultados test gestionar programas

- Pruebas de aceptación

Tabla 45 Caso de prueba registro de facultad

CASO DE PRUEBA	CP-039				
	HU16				
	19/04/2021				
<b>Nombre</b> Registro de facultad					
<b>Descripción</b> El usuario registra una facultad de la UFPS.					
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>					
<b>Resultado esperado</b> Registro exitoso					
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa en la interfaz de facultades cuya url es (host/facultades)</li> <li>2. Dar click en el botón  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario</li> </ol> <table border="1" data-bbox="407 1010 1255 1087"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>	Campo	Valor	Facultad	Ingeniería	
Campo	Valor				
Facultad	Ingeniería				

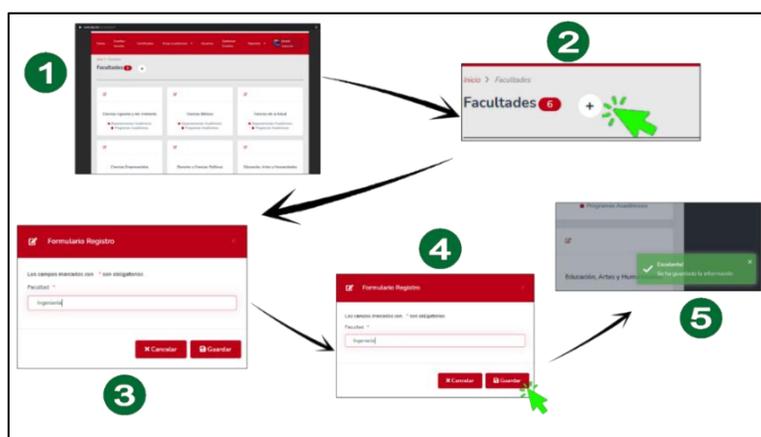


Figura 45 Desarrollo caso de prueba registrar facultad

Tabla 46 Caso de prueba registro de facultad ya existente

CASO DE PRUEBA	CP-040				
	HU16				
	19/04/2021				
<b>Nombre</b> Registro de facultad ya existente					
<b>Descripción</b> El usuario con rol super-admin registra una facultad que ya se registró antes					
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>					
<b>Resultado esperado</b> Registro fallido					
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa en la interfaz de facultades cuya url es (host/facultades)</li> <li>2. Dar click en el botón  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario</li> </ol> <table border="1" data-bbox="407 936 1255 1010"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro fallido</li> </ol>	Campo	Valor	Facultad	Ingeniería	
Campo	Valor				
Facultad	Ingeniería				

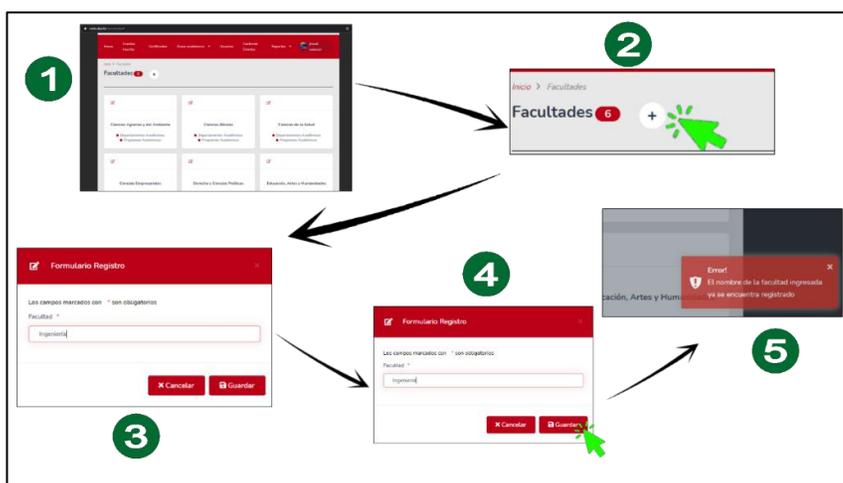


Figura 46 Resultado caso de prueba registro de facultad ya existente.

Tabla 47 Caso de prueba actualización de facultad

CASO DE PRUEBA	CP-041						
	HU16						
	19/04/2021						
<b>Nombre</b> Actualización de facultad							
<b>Descripción</b> El usuario actualiza el nombre de una facultad.							
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>							
<b>Resultado esperado</b> Actualización exitosa							
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. El usuario ingresa en la interfaz de facultades cuya url es (host/facultades)</li> <li>2. Dar click en el botón  sobre una facultad registrada para visualizar la información existente en el formulario</li> <li>3. Ingresar los campos que se van a actualizar en el formulario</li> </ol> <table border="1" data-bbox="407 974 1308 1083"> <thead> <tr> <th>Campo</th> <th>Valor Antiguo</th> <th>Valor nuevo</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ciencias de la Salud</td> <td>Ciencias empresariales</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de actualización exitosa</li> </ol>	Campo	Valor Antiguo	Valor nuevo	Facultad	Ciencias de la Salud	Ciencias empresariales	
Campo	Valor Antiguo	Valor nuevo					
Facultad	Ciencias de la Salud	Ciencias empresariales					

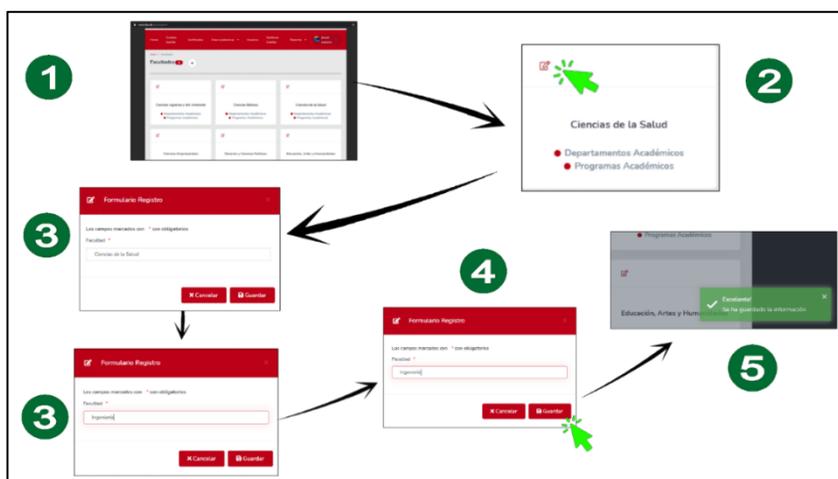


Figura 47 Resultado caso de prueba actualización facultad.

Tabla 48 Caso de prueba registro de departamento académico

CASO DE PRUEBA	CP-042						
	HU16						
	19/04/2021						
<b>Nombre</b> Registro de departamento académico							
<b>Descripción</b> El usuario registra un departamento académico de la UFPS.							
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>							
<b>Resultado esperado</b> Registro exitoso							
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de departamentos académicos</li> <li>2. Dar click en el botón  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario</li> </ol> <table border="1" data-bbox="407 936 1255 1050"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> </tr> <tr> <td>Departamento</td> <td>Sistemas e Informática</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>	Campo	Valor	Facultad	Ingeniería	Departamento	Sistemas e Informática	
Campo	Valor						
Facultad	Ingeniería						
Departamento	Sistemas e Informática						

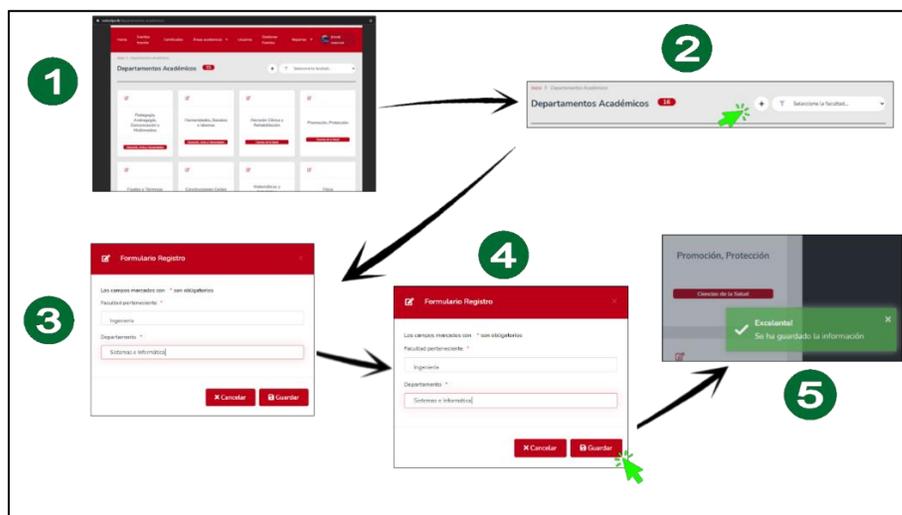


Figura 48. Registro caso de prueba registro de departamento académico.

Tabla 49 Caso de prueba registro de departamento académico ya existente

CASO DE PRUEBA	CP-043						
	HU16						
	19/04/2021						
<b>Nombre</b> Registro de departamento académico ya existente							
<b>Descripción</b> El usuario registra un departamento académico que ya se registró antes							
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>							
<b>Resultado esperado</b> Registro fallido							
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de departamentos académicos</li> <li>2. Dar click en el botón  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario</li> </ol> <table border="1" data-bbox="407 936 1255 1050"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> </tr> <tr> <td>Departamento</td> <td>Sistemas e Informática</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro fallido</li> </ol>	Campo	Valor	Facultad	Ingeniería	Departamento	Sistemas e Informática	
Campo	Valor						
Facultad	Ingeniería						
Departamento	Sistemas e Informática						

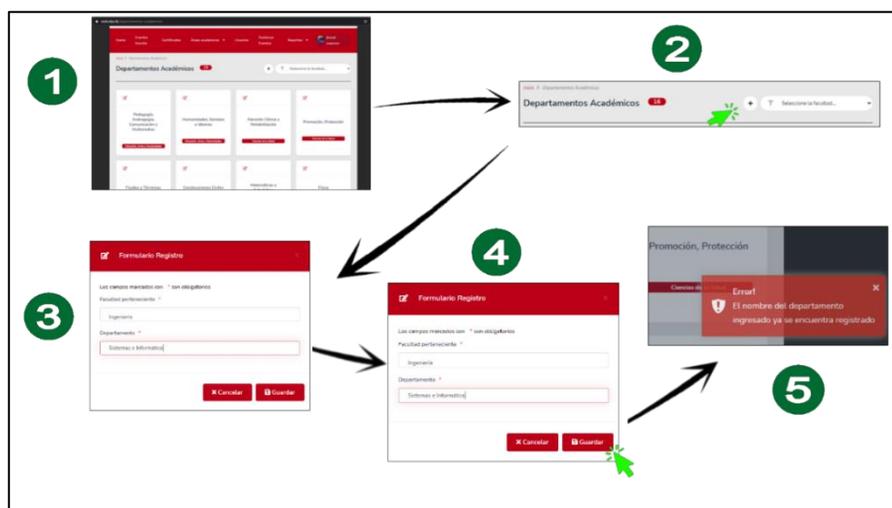


Figura 49 Resultado caso de prueba registro de departamento académico ya existente.

Tabla 50 Caso de prueba actualización de departamento académico

CASO DE PRUEBA	CP-044									
	HU16									
	19/04/2021									
<b>Nombre</b> Actualización de departamento académico										
<b>Descripción</b> El usuario actualiza los datos de un departamento académico correctamente.										
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>										
<b>Resultado esperado</b> Actualización exitosa										
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de departamentos académicos</li> <li>2. Dar click en el botón  sobre un departamento registrado para visualizar la información existente en el formulario</li> <li>3. Ingresar los campos que se van a actualizar en el formulario</li> </ol> <table border="1" data-bbox="407 972 1308 1119"> <thead> <tr> <th>Campo</th> <th>Valor Antiguo</th> <th>Valor nuevo</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> <td>Ciencias empresariales</td> </tr> <tr> <td>Departamento</td> <td>Sistemas e Informática</td> <td>Fluidos y Térmicas</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>	Campo	Valor Antiguo	Valor nuevo	Facultad	Ingeniería	Ciencias empresariales	Departamento	Sistemas e Informática	Fluidos y Térmicas	
Campo	Valor Antiguo	Valor nuevo								
Facultad	Ingeniería	Ciencias empresariales								
Departamento	Sistemas e Informática	Fluidos y Térmicas								

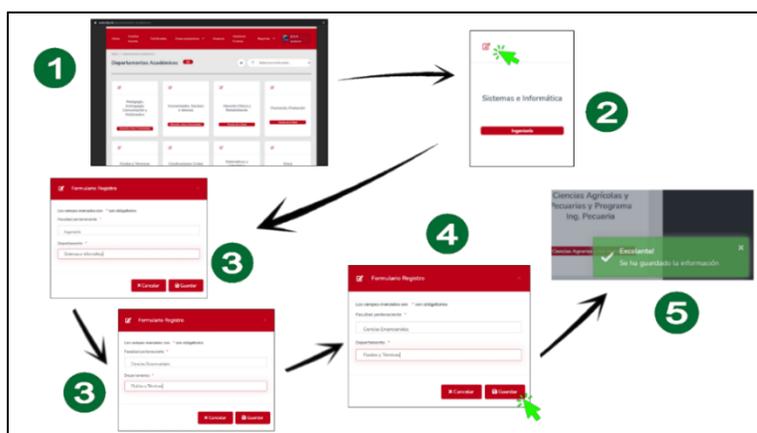


Figura 50 Resultado caso de prueba actualización de departamento académico

Tabla 51 Caso de prueba registro de programa académico

CASO DE PRUEBA		CP-045										
		HU16										
		19/04/2021										
<b>Nombre</b> Registro de programa académico												
<b>Descripción</b> El usuario registra un programa académico de la UFPS.												
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>												
<b>Resultado esperado</b> Registro exitoso												
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de programas académicos</li> <li>2. Dar click en el botón  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario</li> </ol> <table border="1" data-bbox="407 936 1291 1123"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> </tr> <tr> <td>Código</td> <td>115</td> </tr> <tr> <td>Programa</td> <td>Ingeniería de Sistemas</td> </tr> <tr> <td>Director(a)</td> <td>Judith del Pilar Rodríguez Tenjo</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>			Campo	Valor	Facultad	Ingeniería	Código	115	Programa	Ingeniería de Sistemas	Director(a)	Judith del Pilar Rodríguez Tenjo
Campo	Valor											
Facultad	Ingeniería											
Código	115											
Programa	Ingeniería de Sistemas											
Director(a)	Judith del Pilar Rodríguez Tenjo											

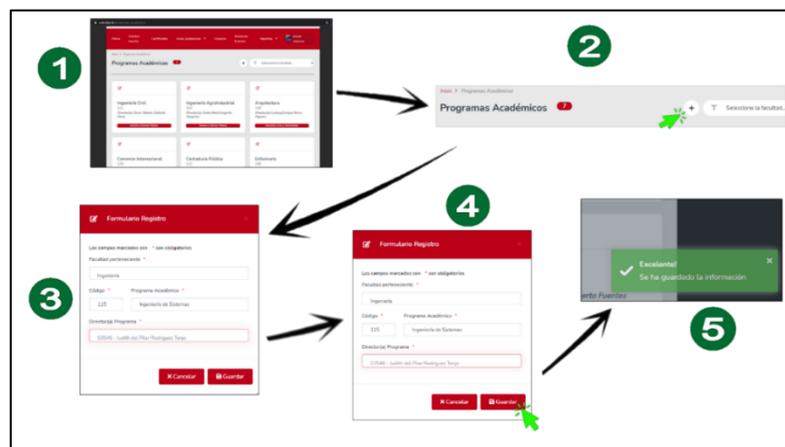


Figura 51 Resultado caso de prueba registro de programa académico.

Tabla 52 Caso de prueba registro de programa académico ya existente

CASO DE PRUEBA	CP-046										
	HU16										
	19/04/2021										
<b>Nombre</b> Registro de programa académico ya existente											
<b>Descripción</b> El usuario registra un programa académico que ya se registró antes											
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>											
<b>Resultado esperado</b> Registro fallido											
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de programas académicos</li> <li>2. Dar click en el botón  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario</li> </ol> <table border="1" data-bbox="407 934 1291 1123"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> </tr> <tr> <td>Código</td> <td>115</td> </tr> <tr> <td>Programa</td> <td>Ingeniería de Sistemas</td> </tr> <tr> <td>Director(a)</td> <td>Judith del Pilar Rodríguez Tenjo</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro fallido</li> </ol>	Campo	Valor	Facultad	Ingeniería	Código	115	Programa	Ingeniería de Sistemas	Director(a)	Judith del Pilar Rodríguez Tenjo	
Campo	Valor										
Facultad	Ingeniería										
Código	115										
Programa	Ingeniería de Sistemas										
Director(a)	Judith del Pilar Rodríguez Tenjo										

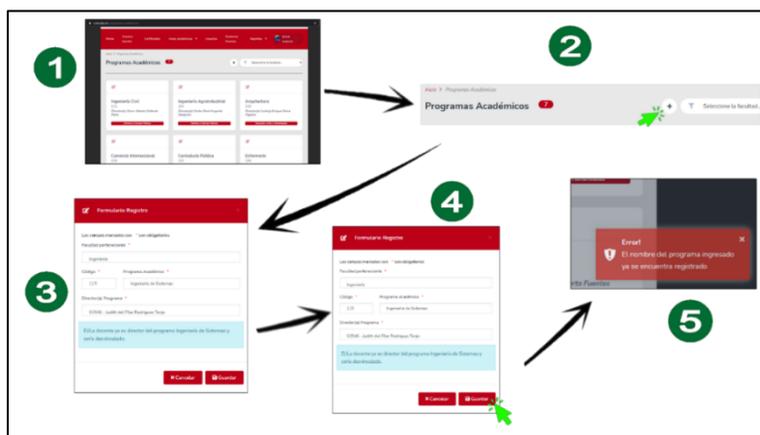


Figura 52 Resultado caso de prueba registro de programa académico ya existente.

Tabla 53 Caso de prueba actualización de programa académico

CASO DE PRUEBA	CP-047															
	HU16															
	19/04/2021															
<b>Nombre</b> Actualización de programa académico																
<b>Descripción</b> El usuario actualiza los datos de un programa académico correctamente.																
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin</li> </ul>																
<b>Resultado esperado</b> Actualización exitosa																
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de programas académicos</li> <li>2. Dar click en el botón  sobre una facultad registrada para visualizar la información existente en el formulario</li> <li>3. Ingresar los campos que se van a actualizar en el formulario</li> </ol> <table border="1" data-bbox="407 974 1308 1241"> <thead> <tr> <th>Campo</th> <th>Valor Antiguo</th> <th>Valor nuevo</th> </tr> </thead> <tbody> <tr> <td>Facultad</td> <td>Ingeniería</td> <td>Ingeniería</td> </tr> <tr> <td>Código</td> <td>115</td> <td>162</td> </tr> <tr> <td>Programa</td> <td>Ingeniería de Sistemas</td> <td>Ingeniería Agronómica</td> </tr> <tr> <td>Director(a)</td> <td>Judith del Pilar Rodríguez Tenjo</td> <td>Jessica Lorena Leal Pabón</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>	Campo	Valor Antiguo	Valor nuevo	Facultad	Ingeniería	Ingeniería	Código	115	162	Programa	Ingeniería de Sistemas	Ingeniería Agronómica	Director(a)	Judith del Pilar Rodríguez Tenjo	Jessica Lorena Leal Pabón	
Campo	Valor Antiguo	Valor nuevo														
Facultad	Ingeniería	Ingeniería														
Código	115	162														
Programa	Ingeniería de Sistemas	Ingeniería Agronómica														
Director(a)	Judith del Pilar Rodríguez Tenjo	Jessica Lorena Leal Pabón														

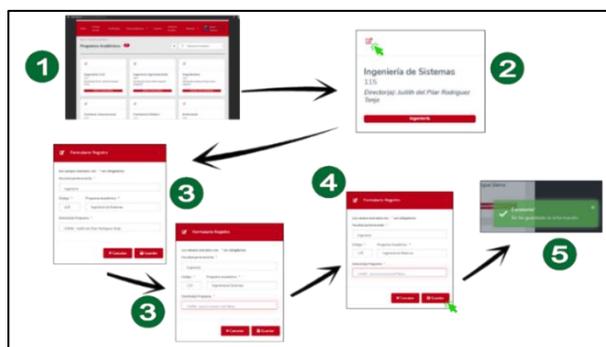
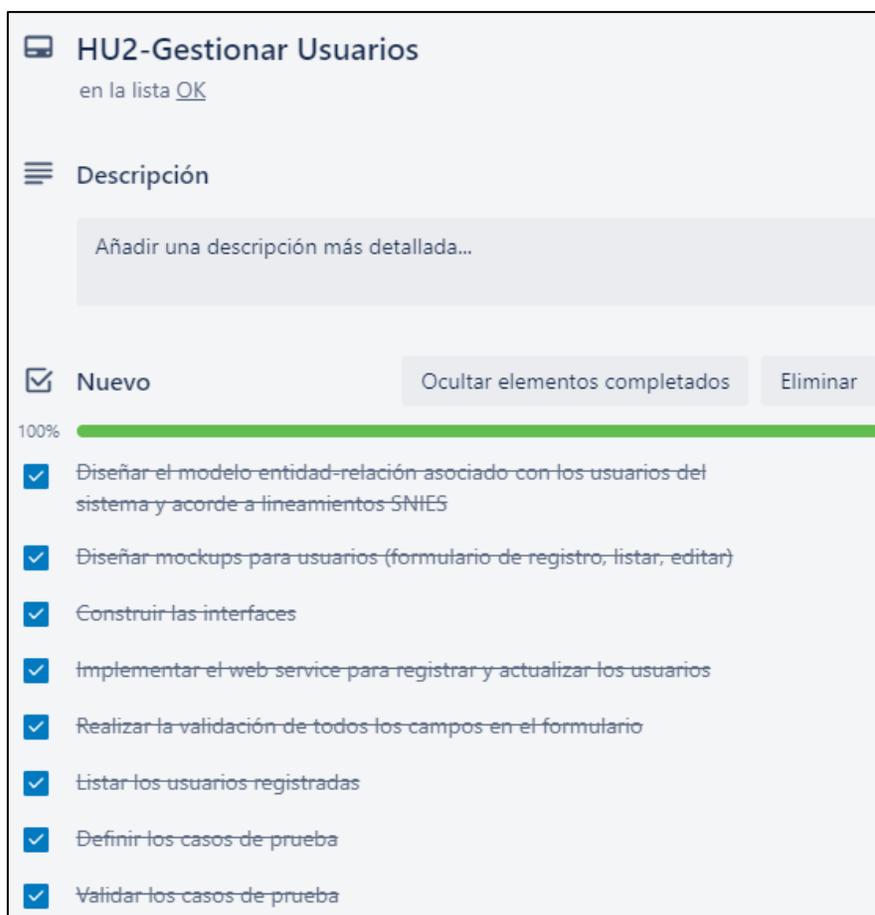


Figura 53 Resultado caso de prueba actualización de programa académico

## ANEXO 1-2. HU2 Gestionar Usuarios

## Planeación de tareas en tarjeta Kanban



**HU2-Gestionar Usuarios**  
en la lista [OK](#)

**Descripción**  
Añadir una descripción más detallada...

**Nuevo** Ocultar elementos completados Eliminar

100%

- Diseñar el modelo entidad-relación asociado con los usuarios del sistema y acorde a lineamientos SNIES
- Diseñar mockups para usuarios (formulario de registro, listar, editar)
- Construir las interfaces
- Implementar el web service para registrar y actualizar los usuarios
- Realizar la validación de todos los campos en el formulario
- Listar los usuarios registradas
- Definir los casos de prueba
- Validar los casos de prueba

Figura 54. Tarjeta Kanban gestionar usuarios

## Definición del modelo de datos

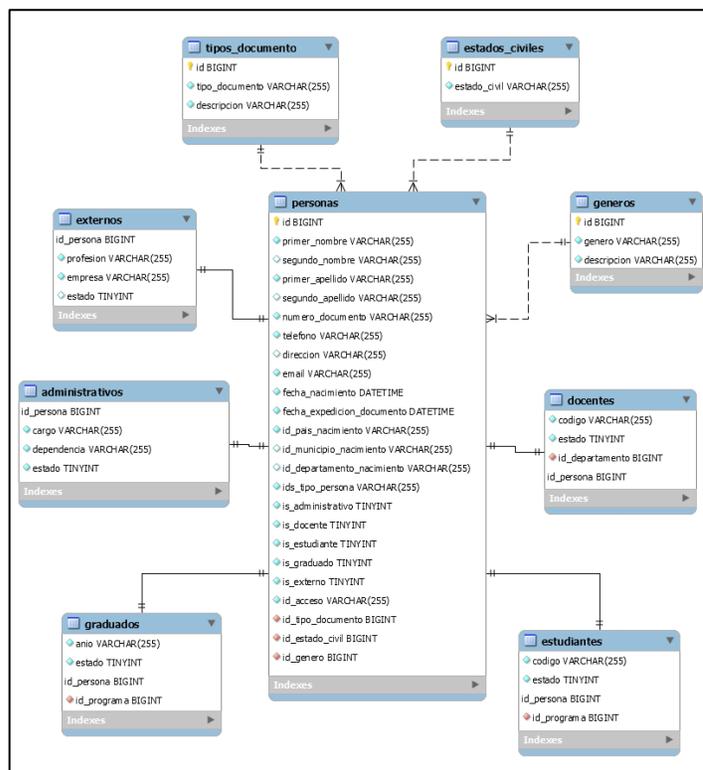


Figura 55 Modelo datos usuarios

## Diseño de mockups

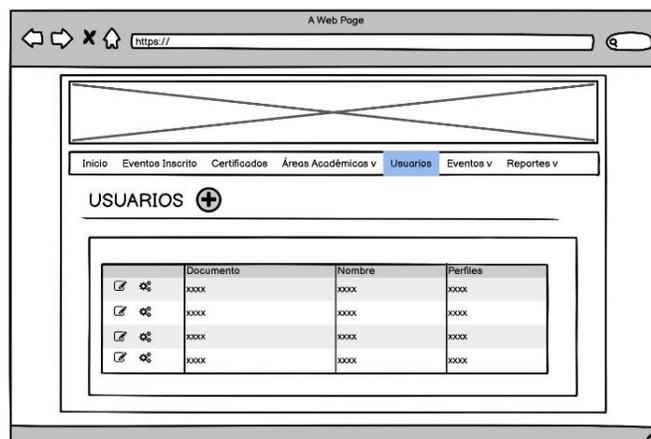


Figura 56 Mockup listar usuarios

Figura 57 Mockup registro usuario.

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IPersonaService personaService;

@Mock
private IPersonaDao personaDao;

@InjectMocks
private PersonaRestController personaController;

private static UsuarioDto USUARIO_ESTUDIANTE;
private static UsuarioDto USUARIO_DOCENTE;
private static UsuarioDto USUARIO_ADMINISTRATIVO;
private static UsuarioDto USUARIO_GRADUADO;
private static UsuarioDto USUARIO_EXTERNO;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    USUARIO_ESTUDIANTE= new UsuarioDto(1L, 1L, "CC", "515194", new Date(),
        "Carla", "Valentina", "Perez", "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(),
        "57", "Colombia", "54", "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null,
        "464484", null, null, true, false, false, false, false, "1548779", 1L, "Ingeniería de Sistemas",
        null, null, null, null, null, null, null);
    USUARIO_DOCENTE= new UsuarioDto(1L, 1L, "CC", "515194", new Date(), "Carla", "Valentina", "Perez",
        "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(), "57", "Colombia", "54",
        "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null, "464484",
        null, null, false, true, false, false, false, null, null, null, null, null, null, 1L,
        "Sistemas e Informática", "016158", null, null);
}

```

Figura 58 Declaración variables test gestionar

```

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    USUARIO_ESTUDIANTE= new UsuarioDto(1L, 1L, "CC", "515194", new Date(),
    "Carla", "Valentina", "Perez", "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(),
    "57", "Colombia", "54", "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null,
    "464484", null, null, true, false, false, false, false, "1548779", 1L, "Ingeniería de Sistemas",
    null, null, null, null, null, null);
    USUARIO_DOCENTE= new UsuarioDto(1L, 1L, "CC", "515194", new Date(), "Carla", "Valentina", "Perez",
    "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(), "57", "Colombia", "54",
    "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null, "464484",
    null, null, false, true, false, false, false, null, null, null, null, null, null, 1L,
    "Sistemas e Informática", "016158", null, null);
    USUARIO_ADMINISTRATIVO= new UsuarioDto(1L, 1L, "CC", "515194", new Date(), "Carla", "Valentina", "Perez",
    "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(), "57", "Colombia", "54",
    "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null, "464484", null, null,
    false, false, true, false, false, null, null,
    "Registro y Control");
    USUARIO_GRADUADO= new UsuarioDto(1L, 1L, "CC", "515194", new Date(), "Carla", "Valentina", "Perez",
    "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(), "57", "Colombia", "54",
    "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null, "464484", null,
    null, false, false, false, true, false, null, null, null, 2L, "Ingeniería Civil", "2013",
    null, null, null, null, null);
    USUARIO_EXTERNO= new UsuarioDto(1L, 1L, "CC", "515194", new Date(), "Carla", "Valentina", "Perez",
    "Quintero", 2L, "Femenino", 1L, "Soltero(a)", new Date(), "57", "Colombia", "54",
    "Norte de Santander", "001", "Cucuta", "carlavalentina@gmail.com", null, "464484", "CENS",
    "Ingeniero Eléctrico", false, false, false, false, true, null, null, null, null,
    null, null, null, null, null, null);
}

```

Figura 60 Inicialización variables test gestionar

```

@Test
public void saveEstudiante() {
    Mockito.doNothing().when(personaService).guardar(USUARIO_ESTUDIANTE);
    final ResponseEntity<?> response = personaController.nuevoUsuario(USUARIO_ESTUDIANTE);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void saveDocente() {
    Mockito.doNothing().when(personaService).guardar(USUARIO_DOCENTE);
    final ResponseEntity<?> response = personaController.nuevoUsuario(USUARIO_DOCENTE);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void saveAdministrativo() {
    Mockito.doNothing().when(personaService).guardar(USUARIO_ADMINISTRATIVO);
    final ResponseEntity<?> response = personaController.nuevoUsuario(USUARIO_ADMINISTRATIVO);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void saveGraduado() {
    Mockito.doNothing().when(personaService).guardar(USUARIO_GRADUADO);
    final ResponseEntity<?> response = personaController.nuevoUsuario(USUARIO_GRADUADO);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void saveExterno() {
    Mockito.doNothing().when(personaService).guardar(USUARIO_EXTERNO);
    final ResponseEntity<?> response = personaController.nuevoUsuario(USUARIO_EXTERNO);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 59 Test guardar diferentes usuarios.

- Pruebas de aceptación

Tabla 54 Caso de prueba registro exitoso de un usuario en el sistema

<b>CASO DE PRUEBA</b>	<b>CP-005</b>																																								
	<b>HU2</b>																																								
	<b>19/04/2021</b>																																								
<b>Nombre</b> Registro exitoso de un usuario en el sistema																																									
<b>Descripción</b> El usuario nuevo se registra diligenciando el formulario correctamente																																									
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• No aplica</li> </ul>																																									
<b>Resultado esperado</b> Registro exitoso																																									
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ingresar a la página principal de la aplicación</li> <li>2. Dar click en registrarse para desplegar el formulario</li> <li>3. Ingresar los campos del formulario, los campos marcados con * son requeridos <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Campo</th> <th style="text-align: center;">Valor</th> </tr> </thead> <tbody> <tr> <td>Tipo Documento *</td> <td>CC – Cedula de Ciudadanía</td> </tr> <tr> <td>Número Documento *</td> <td>699778451</td> </tr> <tr> <td>Fecha Expedición Documento *</td> <td>03-07-1995</td> </tr> <tr> <td>Primer Nombre *</td> <td>María</td> </tr> <tr> <td>Segundo Nombre</td> <td>Camila</td> </tr> <tr> <td>Primer Apellido *</td> <td>Sandoval</td> </tr> <tr> <td>Segundo Apellido *</td> <td>Ortega</td> </tr> <tr> <td>Género *</td> <td>Femenino</td> </tr> <tr> <td>Estado Civil *</td> <td>Soltera</td> </tr> <tr> <td>Fecha Nacimiento *</td> <td>10/02/1970</td> </tr> <tr> <td>País Nacimiento *</td> <td>Colombia</td> </tr> <tr> <td>Departamento Nacimiento</td> <td>Norte de Santander</td> </tr> <tr> <td>Municipio Nacimiento</td> <td>Cúcuta</td> </tr> <tr> <td>Perfiles *</td> <td>Estudiante</td> </tr> <tr> <td>Email *</td> <td>mariacamilaso@ufps.edu.co</td> </tr> <tr> <td>Teléfono *</td> <td>316487495</td> </tr> <tr> <td>Dirección</td> <td>Av 1e 3-54 los pinos</td> </tr> <tr> <td>Programa Académico *</td> <td>Ingeniería Civil</td> </tr> <tr> <td>Código *</td> <td>1113654</td> </tr> </tbody> </table> </li> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>		Campo	Valor	Tipo Documento *	CC – Cedula de Ciudadanía	Número Documento *	699778451	Fecha Expedición Documento *	03-07-1995	Primer Nombre *	María	Segundo Nombre	Camila	Primer Apellido *	Sandoval	Segundo Apellido *	Ortega	Género *	Femenino	Estado Civil *	Soltera	Fecha Nacimiento *	10/02/1970	País Nacimiento *	Colombia	Departamento Nacimiento	Norte de Santander	Municipio Nacimiento	Cúcuta	Perfiles *	Estudiante	Email *	mariacamilaso@ufps.edu.co	Teléfono *	316487495	Dirección	Av 1e 3-54 los pinos	Programa Académico *	Ingeniería Civil	Código *	1113654
Campo	Valor																																								
Tipo Documento *	CC – Cedula de Ciudadanía																																								
Número Documento *	699778451																																								
Fecha Expedición Documento *	03-07-1995																																								
Primer Nombre *	María																																								
Segundo Nombre	Camila																																								
Primer Apellido *	Sandoval																																								
Segundo Apellido *	Ortega																																								
Género *	Femenino																																								
Estado Civil *	Soltera																																								
Fecha Nacimiento *	10/02/1970																																								
País Nacimiento *	Colombia																																								
Departamento Nacimiento	Norte de Santander																																								
Municipio Nacimiento	Cúcuta																																								
Perfiles *	Estudiante																																								
Email *	mariacamilaso@ufps.edu.co																																								
Teléfono *	316487495																																								
Dirección	Av 1e 3-54 los pinos																																								
Programa Académico *	Ingeniería Civil																																								
Código *	1113654																																								

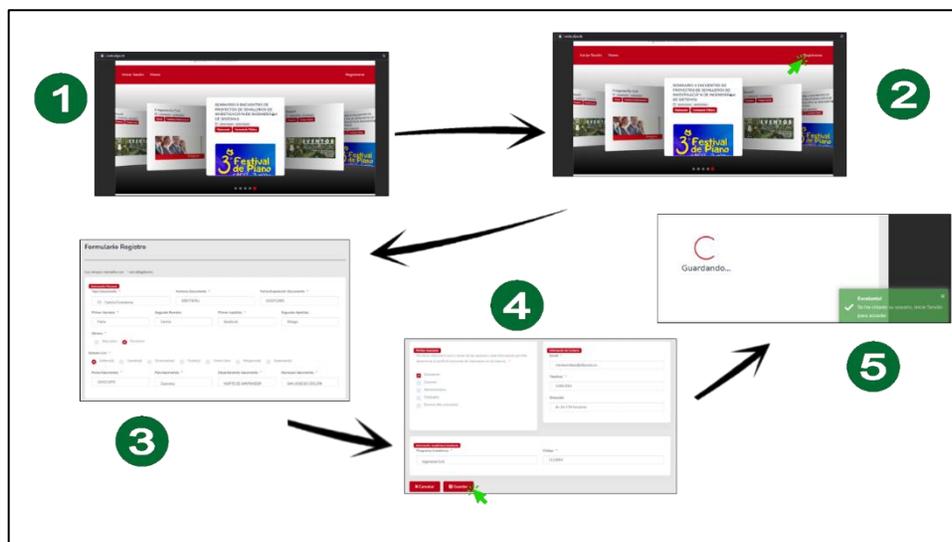


Figura 61 Resultado caso de prueba registro exitoso de un usuario en el sistema.

Tabla 55 Caso de prueba registro fallido de un usuario en el sistema

CASO DE PRUEBA		CP-006
		HU2
		19/04/2021
<b>Nombre</b>	Registro fallido de un usuario en el sistema	
<b>Descripción</b>	El usuario nuevo diligencia el formulario dejando unos campos requeridos sin diligenciar	
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>No aplica</li> </ul>	
<b>Resultado esperado</b>	Registro fallido	
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>Ingresar a la página principal de la aplicación</li> <li>Dar click en “Registrarse” para desplegar el formulario</li> <li>Ingresar los campos del formulario, los campos marcados con * son requeridos</li> </ol>	
	<b>Campo</b>	<b>Valor</b>
	Tipo Documento *	CC – Cedula de Ciudadanía
	Número Documento *	
	Fecha Expedición Documento *	03-07-1995
	Primer Nombre *	Carlos
	Segundo Nombre	

Primer Apellido *	Ramirez
Segundo Apellido *	Perez
Género *	Masculino
Estado Civil *	Soltero
Fecha Nacimiento *	
País Nacimiento *	Colombia
Departamento Nacimiento	Norte de Santander
Municipio Nacimiento	Cúcuta
Perfiles *	Docente
Email *	
Teléfono *	3216748549
Dirección	Av 3e 15-36 Caobos
Departamento Académico *	Matemáticas y Estadística
Código *	064949

4. Dar click en el botón “Guardar”

5. Visualizar el mensaje de registro fallido

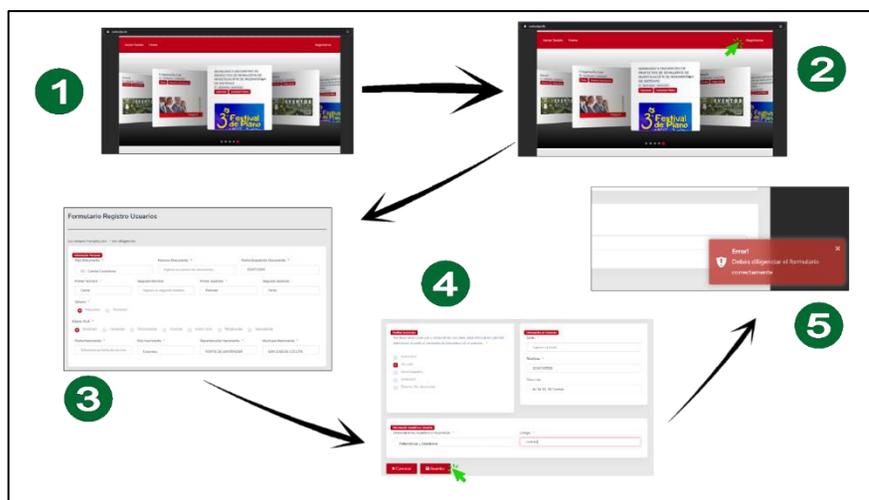


Figura 62 Resultado caso de prueba registro fallido de un usuario en el

Tabla 56 Caso de prueba registro exitoso de un usuario por parte de otro usuario

<b>CASO DE PRUEBA</b>	<b>CP-007</b>
	<b>HU2</b>
	<b>19/04/2021</b>
<b>Nombre</b>	

Registro exitoso de un usuario por parte de otro usuario																																									
<b>Descripción</b> El usuario intenta registrar un usuario diligenciando el formulario correctamente																																									
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANPEOPLE</li> </ul>																																									
<b>Resultado esperado</b> Registro exitoso																																									
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de usuarios</li> <li>2. Dar click en  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario, los campos marcados con * son requeridos</li> </ol> <table border="1" data-bbox="406 756 1347 1501"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Tipo Documento *</td> <td>CC – Cedula de Ciudadanía</td> </tr> <tr> <td>Número Documento *</td> <td>1659741863</td> </tr> <tr> <td>Fecha Expedición Documento *</td> <td>03-10-2013</td> </tr> <tr> <td>Primer Nombre *</td> <td>Paula</td> </tr> <tr> <td>Segundo Nombre</td> <td>Andrea</td> </tr> <tr> <td>Primer Apellido *</td> <td>Carvajal</td> </tr> <tr> <td>Segundo Apellido *</td> <td>Bonilla</td> </tr> <tr> <td>Género *</td> <td>Femenino</td> </tr> <tr> <td>Estado Civil *</td> <td>Soltera</td> </tr> <tr> <td>Fecha Nacimiento *</td> <td>10/08/1995</td> </tr> <tr> <td>País Nacimiento *</td> <td>Colombia</td> </tr> <tr> <td>Departamento Nacimiento</td> <td>Norte de Santander</td> </tr> <tr> <td>Municipio Nacimiento</td> <td>Cúcuta</td> </tr> <tr> <td>Perfiles *</td> <td>Estudiante</td> </tr> <tr> <td>Email *</td> <td>paulandreacb@ufps.edu.co</td> </tr> <tr> <td>Teléfono *</td> <td>314897642</td> </tr> <tr> <td>Dirección</td> <td>Av 1e 3-54 los pinos</td> </tr> <tr> <td>Programa Académico *</td> <td>Arquitectura</td> </tr> <tr> <td>Código *</td> <td>1613548</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>4. Dar click en el botón “Guardar”</li> <li>5. Visualizar el mensaje de registro exitoso</li> </ol>		Campo	Valor	Tipo Documento *	CC – Cedula de Ciudadanía	Número Documento *	1659741863	Fecha Expedición Documento *	03-10-2013	Primer Nombre *	Paula	Segundo Nombre	Andrea	Primer Apellido *	Carvajal	Segundo Apellido *	Bonilla	Género *	Femenino	Estado Civil *	Soltera	Fecha Nacimiento *	10/08/1995	País Nacimiento *	Colombia	Departamento Nacimiento	Norte de Santander	Municipio Nacimiento	Cúcuta	Perfiles *	Estudiante	Email *	paulandreacb@ufps.edu.co	Teléfono *	314897642	Dirección	Av 1e 3-54 los pinos	Programa Académico *	Arquitectura	Código *	1613548
Campo	Valor																																								
Tipo Documento *	CC – Cedula de Ciudadanía																																								
Número Documento *	1659741863																																								
Fecha Expedición Documento *	03-10-2013																																								
Primer Nombre *	Paula																																								
Segundo Nombre	Andrea																																								
Primer Apellido *	Carvajal																																								
Segundo Apellido *	Bonilla																																								
Género *	Femenino																																								
Estado Civil *	Soltera																																								
Fecha Nacimiento *	10/08/1995																																								
País Nacimiento *	Colombia																																								
Departamento Nacimiento	Norte de Santander																																								
Municipio Nacimiento	Cúcuta																																								
Perfiles *	Estudiante																																								
Email *	paulandreacb@ufps.edu.co																																								
Teléfono *	314897642																																								
Dirección	Av 1e 3-54 los pinos																																								
Programa Académico *	Arquitectura																																								
Código *	1613548																																								

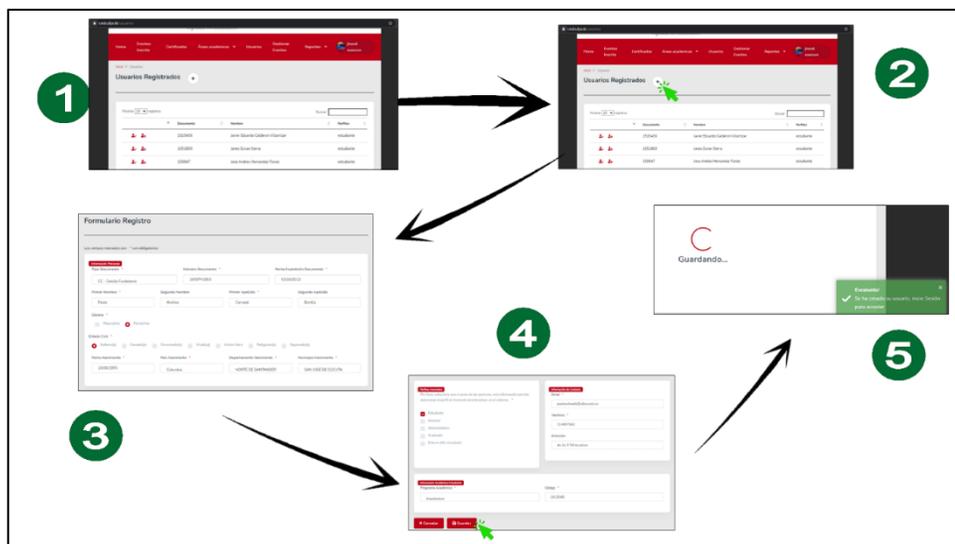


Figura 63 Resultado caso de prueba registro exitoso de un usuario por parte de otro usuario.

Tabla 57 Registro fallido de un usuario con número de documento ya existente

<b>CASO DE PRUEBA</b>	<b>CP-008</b>								
	<b>HU2</b>								
	<b>19/04/2021</b>								
<b>Nombre</b> Registro fallido de un usuario con número de documento ya existente									
<b>Descripción</b> El usuario intenta registrar un usuario con un número de documento que ya fue registrado									
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANPEOPLE</li> </ul>									
<b>Resultado esperado</b> Registro fallido									
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de usuarios</li> <li>2. Dar click en  para desplegar el formulario</li> <li>3. Ingresar los campos del formulario, los campos marcados con * son requeridos</li> </ol> <table border="1" style="width: 100%; margin-top: 10px;"> <thead> <tr> <th style="text-align: center;">Campo</th> <th style="text-align: center;">Valor</th> </tr> </thead> <tbody> <tr> <td>Tipo Documento *</td> <td>CC – Cedula de Ciudadanía</td> </tr> <tr> <td>Número Documento *</td> <td>1659741863</td> </tr> <tr> <td>Fecha Expedición Documento *</td> <td>03-10-2013</td> </tr> </tbody> </table>	Campo	Valor	Tipo Documento *	CC – Cedula de Ciudadanía	Número Documento *	1659741863	Fecha Expedición Documento *	03-10-2013	
Campo	Valor								
Tipo Documento *	CC – Cedula de Ciudadanía								
Número Documento *	1659741863								
Fecha Expedición Documento *	03-10-2013								

Primer Nombre *	Paula
Segundo Nombre	Andrea
Primer Apellido *	Carvajal
Segundo Apellido *	Bonilla
Género *	Femenino
Estado Civil *	Soltera
Fecha Nacimiento *	10/08/1995
País Nacimiento *	Colombia
Departamento Nacimiento	Norte de Santander
Municipio Nacimiento	Cúcuta
Perfiles *	Estudiante
Email *	paulandreacb@ufps.edu.co
Teléfono *	314897642
Dirección	Av 1e 3-54 los pinos
Programa Académico *	Arquitectura
Código *	1613548

4. Dar click en el botón “Guardar”
5. Visualizar el mensaje de registro fallido por documento ya existente

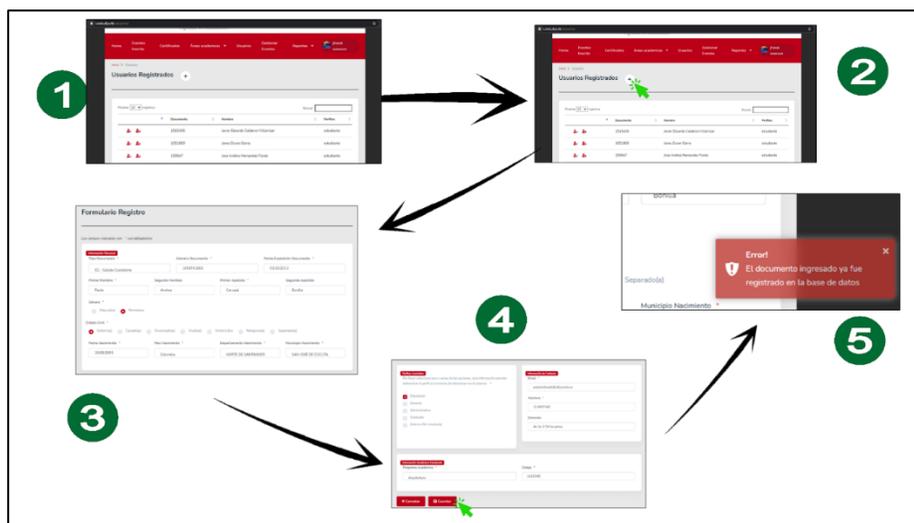


Figura 64 Resultado caso de prueba registro fallido de un usuario con número de documento ya existente.

Tabla 58 Caso de prueba visualizar listado de usuarios

CASO DE PRUEBA	CP-009
	HU2
	19/04/2021
<b>Nombre</b> Visualizar listado de usuarios	
<b>Descripción</b> El usuario visualiza el listado de los usuarios registrados	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>El usuario debe estar logueado</li> <li>El usuario debe tener rol: ROLE_MANPEOPLE</li> </ul>	
<b>Resultado esperado</b> Listado de usuarios	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>Dar click en el menú “Usuarios”</li> <li>Visualizar el listado de los usuarios registrados</li> </ol>	



Figura 65 Resultado caso de prueba visualizar usuarios.

## ANEXO 1-3. HU1 Gestionar eventos y cursos de educación continua

## Planeación de tareas en tarjeta Kanban

HU1-Gestionar cursos y eventos de educación continua  
en la lista [OK](#)

Descripción

Añadir una descripción más detallada...

**Nuevo** Ocultar elementos completados Eliminar

100%

- Diseñar el modelo entidad-relación asociado con la educación continua
- Diseñar mockups para educación continua (formulario de registro; detalles)
- Construir las interfaces
- Implementar el web service para registrar y actualizar la educación continua
- Realizar la validación de todos los campos en el formulario
- Listar las educaciones continuas registradas
- Definir los casos de prueba
- Validar los casos de prueba

Figura 66 Tarjeta Kanban gestionar educaciones continuas.

HU1-Gestionar cursos y eventos de educación continua  
en la lista [Hecho](#)

Descripción

Añadir una descripción más detallada...

**Mejoras** Ocultar elementos completados Eliminar

100%

- Permitir el registro de una educación continua tomando como base una ya existente
- Implementar un loader para evitar múltiples clicks al guardar una educación continua
- Implementar un WYSIWYG HTML Editor para ingresar información adicional de una educación continua (imagenes, videos, listas, url; tablas)

Añada un elemento

Figura 67 Tarjeta Kanban gestionar educaciones continuas

## Definición del modelo de datos

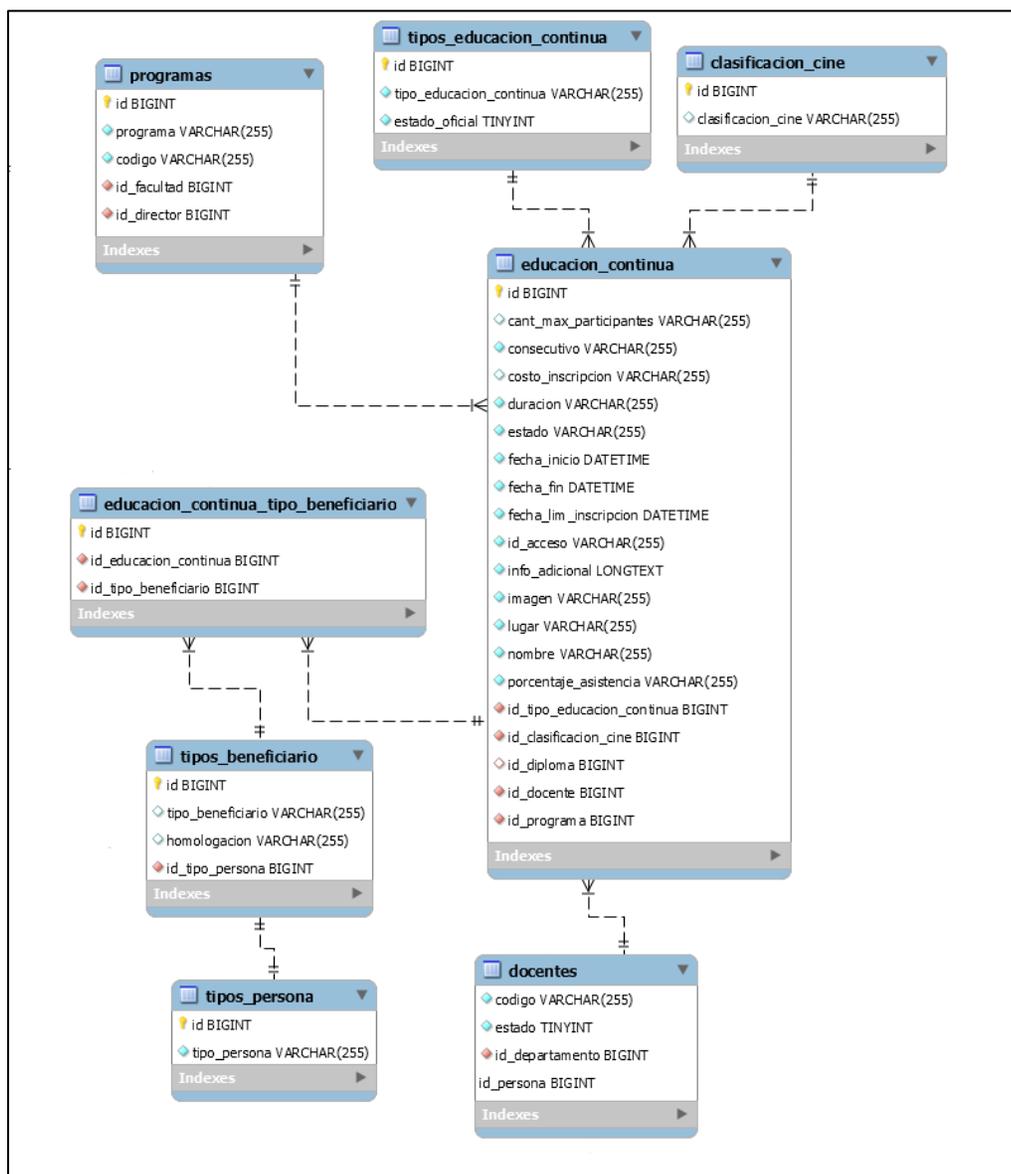


Figura 68 Modelo Entidad-Relación educación continua.

## Diseño de mockups

The mockup shows a web browser window titled "A Web Page" with a navigation menu containing: Inicio, Eventos Inscrito, Certificados, Áreas Académicas v, Usuarios, **Eventos v**, and Reportes v. The main content area is titled "FORMULARIO REGISTRO" and is divided into four sections:

- Información Evento:** Includes a dropdown for "Tipo Educación Continua", a text field for "Número Documento", a dropdown for "Clasificación CINE", text fields for "Fecha Inicio" and "Fecha Fin", a text field for "Duración", a text field for "Costo Educación Continua", and a dropdown for "Imagen" with the label "Seleccione archivo...".
- Información Organizador:** Includes a dropdown for "Programa Académico" and a dropdown for "Docente Responsable".
- Información Participantes:** Includes a text field for "Tipo Beneficiarios", a text field for "Fecha Límite Inscripción", a text field for "Costo Inscripción", a text field for "Cantidad participantes", and a text field for "Porcentaje de asistencia Mínimo".
- Información Adicional:** Includes a text field for "Lugar" and a large text area for "Detalles".

At the bottom of the form are two buttons: "X Cancelar" and "✓ Guardar".

Figura 69 Mockup registro educación continua

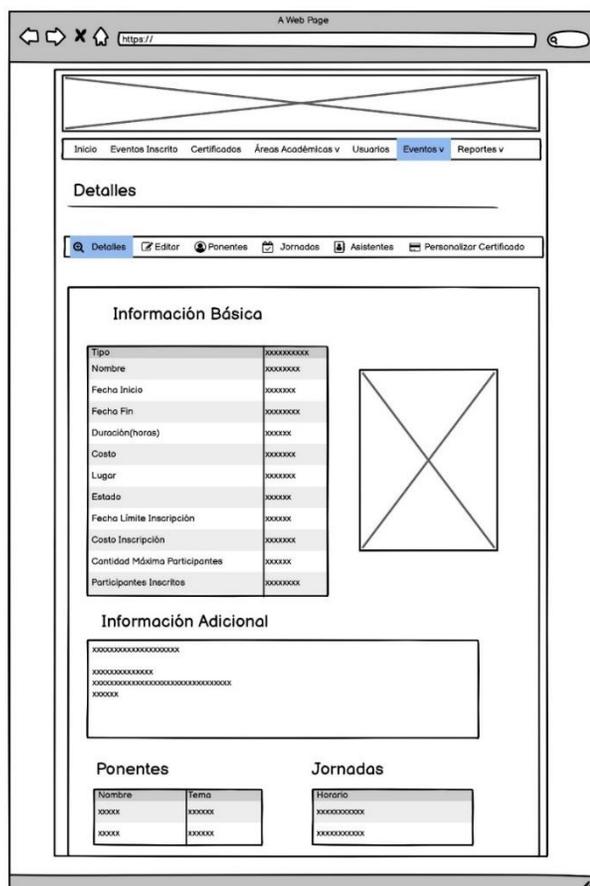


Figura 71 Mockup detalles educación continua

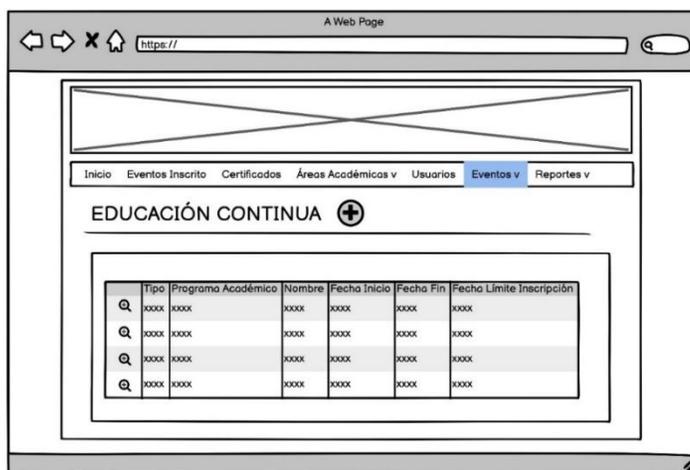


Figura 70 Mockup listar educaciones continuas.

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IEducacionContinuaService educacionContinuaService;

@Mock
private IDiplomaService diplomaService;

@Mock
private IParticipanteService participanteService;

@InjectMocks
private EducacionContinuaRestController educacionContinuaController;

private static EducacionContinuaWebDto EDC_WEB_DTO;

private static List<String> LIST_EDUCACIONES_CONTINUAS;
@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    EDC_WEB_DTO= new EducacionContinuaWebDto();
    EDC_WEB_DTO.setNombre("eisi");
    EDC_WEB_DTO.setCantidadInscritos(20);
    EDC_WEB_DTO.setConsecutivo("001");
    EDC_WEB_DTO.setFechaInicio(new Date());
    EDC_WEB_DTO.setFechaFin(new Date());
    EDC_WEB_DTO.setFechaLimInscripcion(new Date());

    LIST_EDUCACIONES_CONTINUAS=new ArrayList<String>();
    LIST_EDUCACIONES_CONTINUAS.add("eisi");
    LIST_EDUCACIONES_CONTINUAS.add("ciinatic");
    LIST_EDUCACIONES_CONTINUAS.add("semilleros de investigación");
}

```

Figura 72 Instanciación variables gestionar educación continua

```

@Test
public void saveEducacionContinuaTest() {
    final ResponseEntity<?> response = educacionContinuaController.save(null, "1", "eisi",
        "15-02-2021", "18-02-2021", "35", "20", "14-02-2021",
        "35600", "Auditorio JJ Maldonado", "600000", "20", null,
        "1", "Taller", "1", "1", "5", "001", "2");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void searchEducacionContinuaBaseTest() {
    Mockito.when(educacionContinuaService.findEducacionContinuaBase("eisi")).
        thenReturn(EDC_WEB_DTO);
    final ResponseEntity<EducacionContinuaWebDto> response = educacionContinuaController.
        searchEducacionContinuaBase("eisi");
    assertNotNull(response.getBody());
    assertEquals(response.getBody().getCantidadInscritos(), 20);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 73 Test guardar y buscar educación continua

```

@Test
public void searchListEducacionesContinuasByProgramaTest() {
    Mockito.when(educacionContinuaService.findEducacionesContinuasBaseByIdPrograma(Long.parseLong("1")))
        .thenReturn(LIST_EDUCACIONES_CONTINUAS);
    final ResponseEntity<List<String>> response = educacionContinuaController.
        searchListEducacionesContinuasByPrograma("1");
    assertNotNull(response);
    assertEquals(response.getBody().size(), 3);
}

@Test
public void deleteEducacionContinuaTest() {
    final ResponseEntity<?> response = educacionContinuaController.delete("106487987");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 75 Test eliminar y buscar educación continua por programa

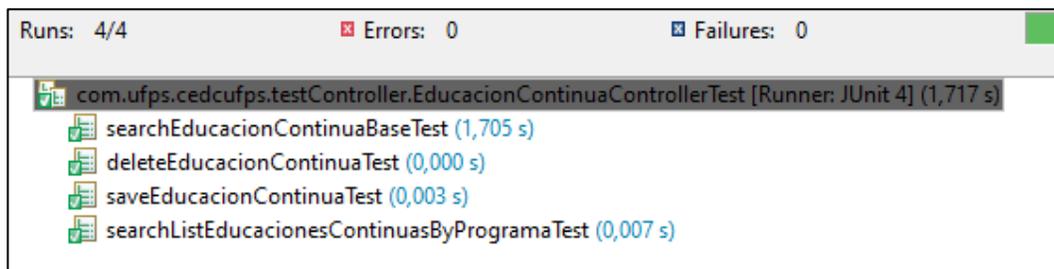


Figura 74 Resultados test gestionar educación continua

- Pruebas de aceptación

Tabla 59 Caso de prueba registro exitoso de curso o evento de educación continua

CASO DE PRUEBA	CP-001
	HU1
	19/04/2021
<b>Nombre</b> Registro exitoso de curso o evento de educación continua	
<b>Descripción</b> El usuario registra un curso o evento de educación continua diligenciando el formulario correctamente	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> </ul>	

- El usuario debe tener rol: ROLE\_MANAECCU

### Resultado esperado

Registro exitoso

### Desarrollo

1. Ubicarse en la interfaz de gestionar eventos
2. Dar click en el botón  para agregar una nueva educación continua
3. Ingresar los campos del formulario, los campos marcados con \* son requeridos

Campo	Valor Antiguo
Tipo Educación Continua *	Simposio
Clasificación CINE *	Ciencias de la Educación
Nombre *	Simposio Internacional sobre minería de datos
Fecha Inicio *	17/08/2020 14:00
Fecha Fin *	19/08/2020 12:00
Duración *	20
Imagen *	<<Se selecciona una imagen>>
Programa organizador *	Ingeniería de Sistemas
Docente Responsable *	05848-Marco Antonio Adarme Jaimes
Tipo Beneficiarios *	Estudiantes de las IES
Fecha Limite Inscripción *	15/08/2020 00:00
Costo Educación Continua *	30000
Costo Inscripción	20000
Cantidad máxima estudiante	40
Lugar *	Auditorio JJ Maldonado
Detalles	Este simposio ofrece un foro de encuentro a investigadores que tratan de extraer conocimiento útil a partir de grandes volúmenes de información mediante el uso de métodos de minería de datos. Las técnicas de minería de datos se han convertido en una herramienta fundamental en la investigación científica y técnica como método de análisis y descubrimiento de conocimiento a partir de datos de observación o de resultados de experimentos para la toma de decisiones en el control de procesos productivos, planificación de sistemas, marketing, etc.

Se desarrollará un panel con los participantes donde se discutirán las necesidades y retos de los diferentes sectores en Minería de datos. El foro contará con la presencia de invitados internacionales, algunos proveedores de tecnología y representantes del gobierno

4. Dar Click en Guardar
5. Visualizar el mensaje de registro exitoso

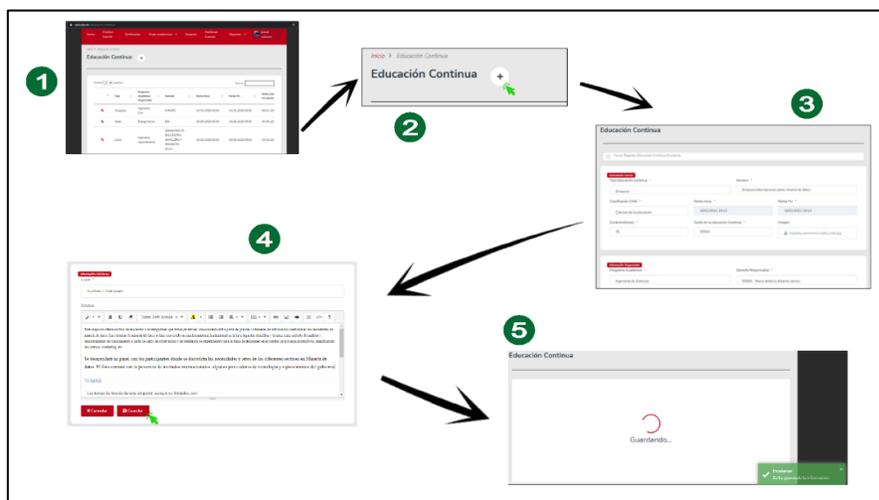


Figura 76 Resultado caso de prueba registro exitoso de curso o evento de educación continua.

Tabla 60 Caso de prueba registro fallido de curso o evento de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-002</b>
	<b>HU1</b>
	<b>19/04/2021</b>
<b>Nombre</b> Registro fallido de curso o evento de educación continua	
<b>Descripción</b> El usuario registra un curso o evento de educación continua diligenciando el formulario incorrectamente	

<b>Prerrequisitos</b>	
<ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>	
<b>Resultado esperado</b>	
Registro fallido	
<b>Desarrollo</b>	
<ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón <input type="button" value="+"/> para agregar una nueva educación continua</li> <li>3. Ingresar los campos del formulario, los campos marcados con * son requeridos</li> </ol>	
Campo	Valor
Tipo Educación Continua *	Simposio
Clasificación CINE *	Ciencias de la Educación
Nombre *	Simposio Internacional sobre minería de datos
Fecha Inicio *	17/08/2020 14:00
Fecha Fin *	19/08/2020 12:00
Duración *	
Imagen *	<<Se selecciona una imagen>>
Programa organizador *	Ingeniería de Sistemas
Docente Responsable *	05848-Marco Antonio Adarme Jaimes
Tipo Beneficiarios *	Estudiantes de las IES
Fecha Limite Inscripción *	15/08/2020 00:00
Costo Educación Continua *	
Costo Inscripción	20000
Cantidad máxima estudiante	40
Lugar *	
Contenido General	Este simposio ofrece un foro de encuentro a investigadores que tratan de extraer conocimiento útil a partir de grandes volúmenes de información mediante el uso de métodos de minería de datos. Las técnicas de minería de datos se han convertido en una herramienta fundamental en la investigación científica y técnica como método de análisis y descubrimiento de conocimiento a partir de datos de observación o de resultados de experimentos para la toma de

decisiones en el control de procesos productivos, planificación de sistemas, marketing, etc. Se desarrollará un panel con los participantes donde se discutirán las necesidades y retos de los diferentes sectores en Minería de datos. El foro contará con la presencia de invitados internacionales, algunos proveedores de tecnología y representantes del gobierno

4. Dar Click en Guardar
5. Visualizar el mensaje de registro fallido

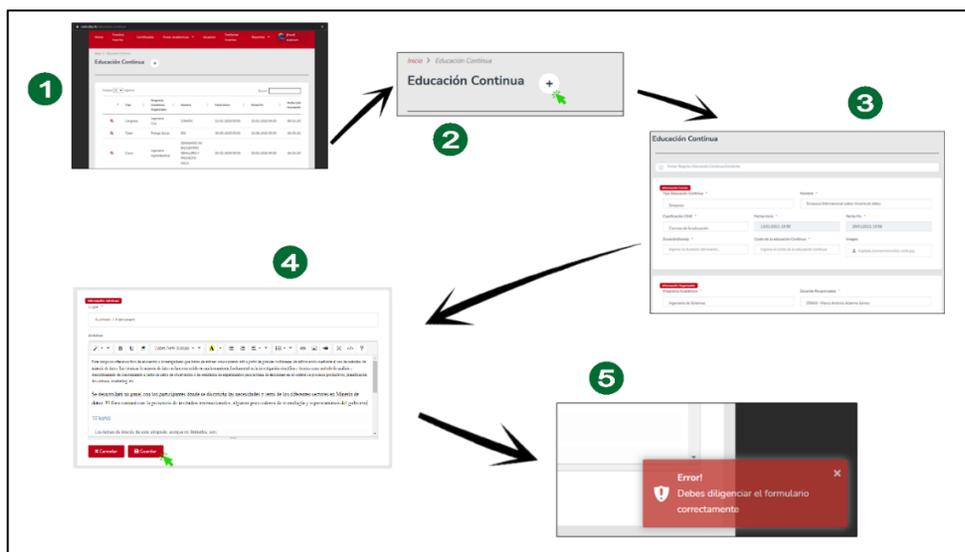


Figura 77 Resultado caso de prueba registro fallido de curso o evento de educación continua

Tabla 61 Caso de prueba visualizar el listado de cursos o eventos de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-003</b>
	<b>HU1</b>
	<b>19/04/2021</b>
<b>Nombre</b>	

Visualizar el listado de cursos o eventos de educación continua
<b>Descripción</b> El usuario visualiza el listado de todos los cursos o eventos de educación continua
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>El usuario debe estar logueado</li> <li>El usuario debe tener rol: ROLE_MANAECCU</li> </ul>
<b>Resultado esperado</b> Listado de los registros solicitados
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>Dar click en el menú “Gestionar Eventos”</li> <li>Visualizar el listado de cursos o eventos de educación continua</li> </ol>

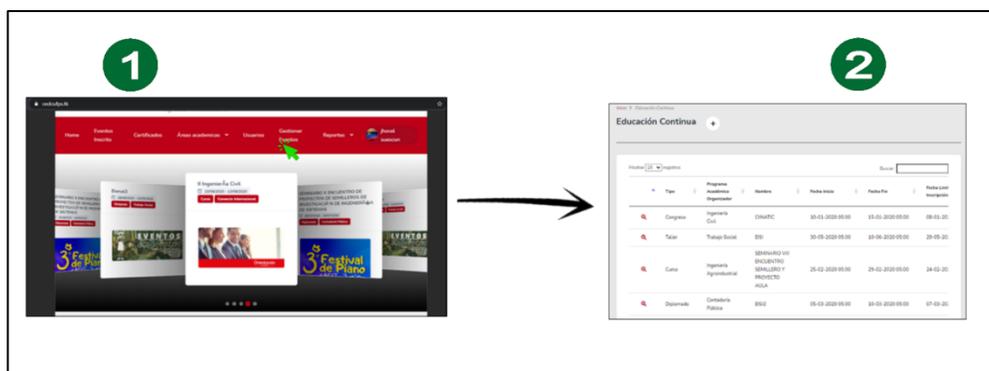


Figura 78 Resultado caso de prueba visualizar el listado de cursos o eventos de educación continua.

Tabla 62 Caso de prueba actualización exitosa de curso o evento de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-004</b>
	<b>HU1</b>
	<b>19/04/2021</b>
<b>Nombre</b> Actualización exitosa de curso o evento de educación continua	
<b>Descripción</b> El usuario edita y actualiza los campos de un curso o evento de educación continua diligenciando el formulario correctamente	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>El usuario debe estar logueado</li> <li>El usuario debe tener rol: ROLE_MANAECCU</li> <li>Debe existir el registro del curso o evento de educación continua que se va a</li> </ul>	

actualizar		
<b>Resultado esperado</b>		
Actualización exitosa		
<b>Desarrollo</b>		
<ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “Editar”</li> <li>4. Ingresar los campos a actualizar, los campos marcados con * son requeridos</li> </ol>		
Campo	Valor Antiguo	Valor Nuevo
Tipo Educación Continua *	Simposio	Taller
Clasificación CINE *	Ciencias de la Educación	Ciencias de la Educación
Nombre *	Simposio Internacional sobre minería de datos	Simposio Internacional sobre minería de datos
Fecha Inicio *	17/08/2020 14:00	17/08/2020 14:00
Fecha Fin *	19/08/2020 12:00	19/08/2020 12:00
Duración *	20	20
Imagen *	<<Se selecciona una imagen>>	<<Se selecciona una imagen>>
Programa organizador *	Ingeniería de Sistemas	Ingeniería de Sistemas
Docente Responsable *	05848-Marco Antonio Adarme Jaimes	05848-Marco Antonio Adarme Jaimes
Tipo Beneficiarios *	Estudiantes de las IES	Estudiantes de las IES
Fecha Limite Inscripción *	15/08/2020 00:00	15/08/2020 00:00
Costo Educación Continua *	30000	30000
Costo Inscripción	20000	20000
Cantidad máxima estudiante	40	30
Lugar *	Auditorio JJ Maldonado	Auditorio Eustorgio Colmenares
Contenido General	Este simposio ofrece un foro de encuentro a investigadores que tratan de extraer conocimiento útil a partir de grandes	

## Resumen

volúmenes de información mediante el uso de métodos de minería de datos. Las técnicas de minería de datos se han convertido en una herramienta fundamental en la investigación científica y técnica como método de análisis y descubrimiento de conocimiento a partir de datos de observación o de resultados de experimentos para la toma de decisiones en el control de procesos productivos, planificación de sistemas, marketing, etc.

Se desarrollará un panel con los participantes donde se discutirán las necesidades y retos de los diferentes sectores en Minería de datos. El foro contará con la presencia de invitados internacionales, algunos proveedores de tecnología y representantes del gobierno

5. Dar Click en Guardar
6. Visualizar el mensaje de actualización exitosa

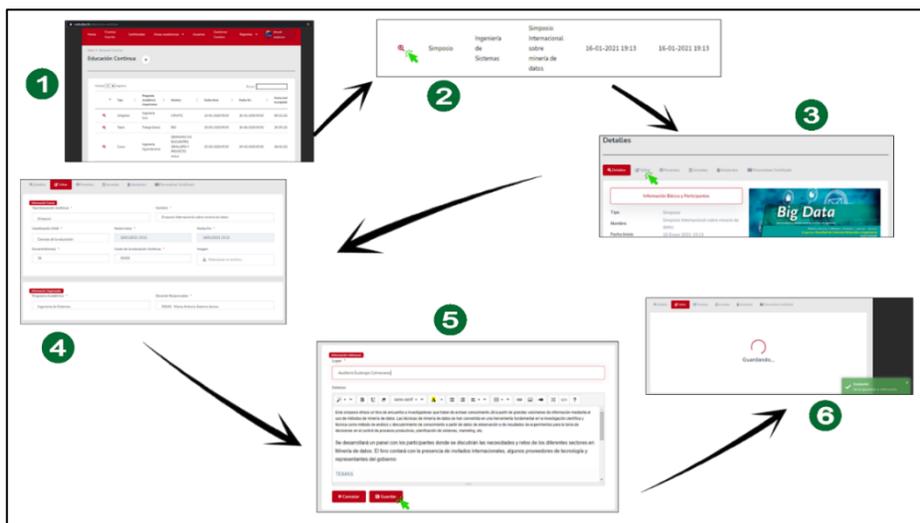


Figura 79 Resultado caso de prueba actualización exitosa de curso o evento de educación continua



## Definición del modelo de datos

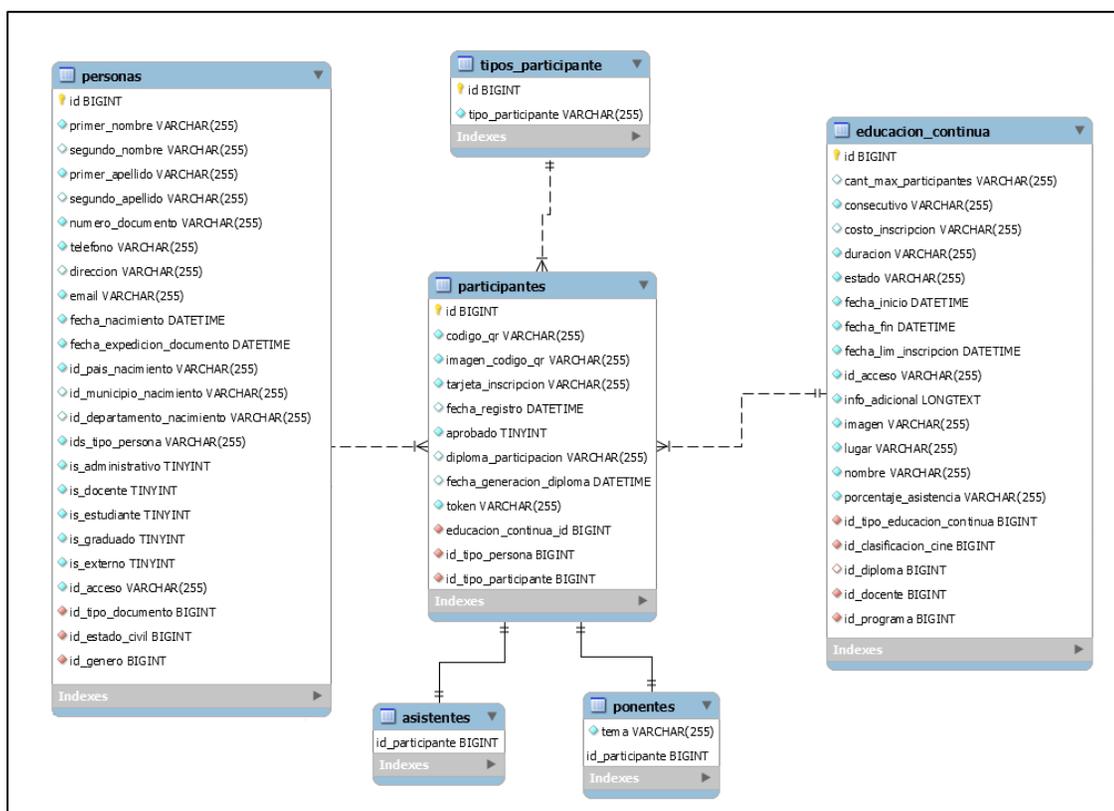


Figura 82 Modelo de datos participantes



## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IParticipanteService participanteService;

@InjectMocks
private AsistenteRestController asistenteRestController;

private static String ID_EDU_CONTINUA;
private static String ID_TIPO_PERSONA;
private static ParticipanteDto PARTICIPANTE_DTO;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    ID_EDU_CONTINUA="1";
    ID_TIPO_PERSONA="1";
    PARTICIPANTE_DTO=new ParticipanteDto();
    PARTICIPANTE_DTO.setId(1L);
    PARTICIPANTE_DTO.setIdEducacionContinua(Long.parseLong(ID_EDU_CONTINUA));
    PARTICIPANTE_DTO.setIdTipoPersona(Long.parseLong(ID_TIPO_PERSONA));
    PARTICIPANTE_DTO.setIdPersona(1L);
}

```

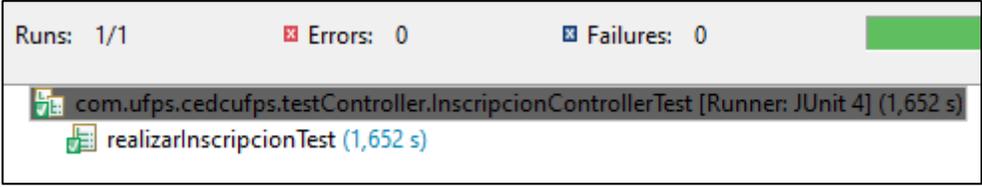
Figura 85. Instanciación variables test realizar inscripción

```

@Test
public void realizarInscripcionTest() {
    Mockito.when(participanteService.saveAsistente(Long.parseLong(ID_EDU_CONTINUA),
        Long.parseLong(ID_TIPO_PERSONA))).thenReturn(PARTICIPANTE_DTO);
    final ResponseEntity<ParticipanteDto> response = asistenteRestController.realizarInscripcion
        (ID_EDU_CONTINUA, ID_TIPO_PERSONA);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
    assertNotNull(response.getBody());
}

```

Figura 86 Test realizar inscripción.



The screenshot shows a test runner interface with the following details:

- Runs: 1/1
- Errors: 0
- Failures: 0
- Test class: com.ufps.cedcufps.testController.InscripcionControllerTest [Runner: JUnit 4] (1,652 s)
- Test method: realizarInscripcionTest (1,652 s)

Figura 87 Resultados test realizar inscripción.

- Pruebas de aceptación

Tabla 63 Caso de prueba inscripción exitosa de un usuario a un curso o evento de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-010</b>
	<b>HU3</b>
	<b>19/04/2021</b>
<b>Nombre</b>	Inscripción exitosa de un usuario a un curso o evento de educación continua
<b>Descripción</b>	El usuario se inscribe a un curso o evento de educación continua disponible
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario no debe haberse inscrito antes</li> <li>• Cumplir con los requisitos de inscripción, tales como, tener un perfil del público al cual se ofertó la educación continua</li> </ul>
<b>Resultado esperado</b>	Inscripción exitosa
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>1. Visualizar el panel de eventos activos</li> <li>2. Dar click en el nombre de educación continua activa para ampliar la información</li> <li>3. ¡Dar click en “inscríbete!”</li> <li>4. Visualizar mensaje de inscripción exitosa</li> </ol>

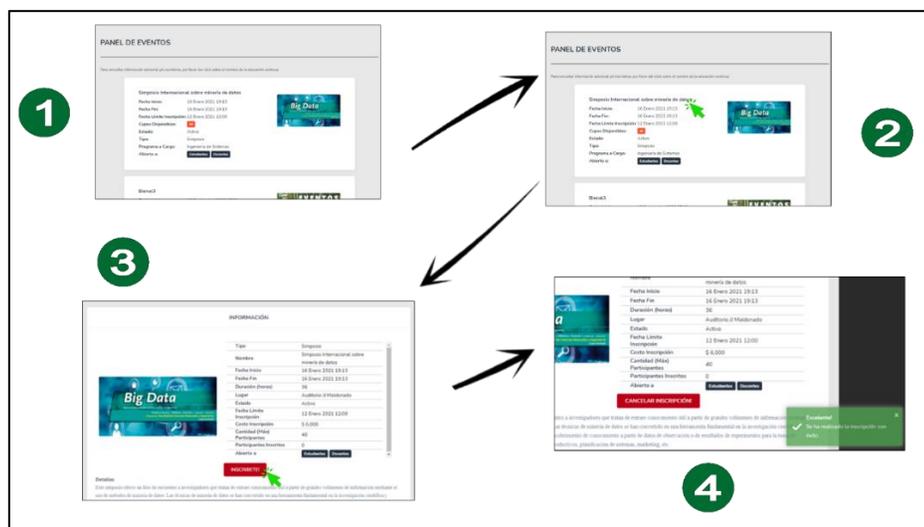


Figura 88 Resultado caso de prueba realizar inscripción.

Tabla 64 Caso de prueba visualización de tarjeta de inscripción

CASO DE PRUEBA	CP-011
	HU3
	19/04/2021
<b>Nombre</b> Visualización de tarjeta de inscripción del curso o evento de educación continua	
<b>Descripción</b> El usuario consulta la tarjeta de inscripción de los cursos o eventos de educación continua en los cuales se ha inscrito	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe estar inscrito a un curso o evento de educación continua</li> <li>• El curso o evento de educación continua debe no tener un estado de terminado</li> </ul>	
<b>Resultado esperado</b> Visualización de QR de inscripción	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Dar click en el menú “Eventos Inscrito”</li> <li>2. Visualizar las inscripciones realizadas de educaciones continuas activas</li> <li>3. Dar click sobre el botón  para visualizar la tarjeta de inscripción de una educación continua</li> <li>4. Visualizar tarjeta de inscripción</li> </ol>	

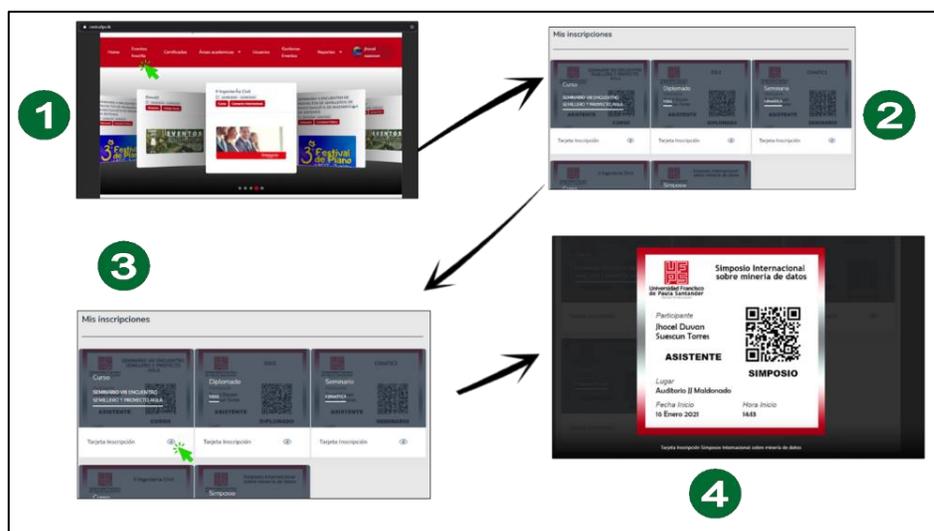


Figura 89 Resultado caso de prueba visualización de tarjeta de

Tabla 65 Caso de prueba inscripción de un usuario a un curso o evento de educación continua sin cumplir requisitos

CASO DE PRUEBA	CP-012
	HU3
	19/04/2021
<b>Nombre</b>	Inscripción de un usuario a un curso o evento de educación continua sin cumplir con el perfil del público al cuál se ofertó
<b>Descripción</b>	El usuario intenta inscribirse a un curso o evento de educación continua activo el cuál fue ofertado a un tipo de público distinto al cuál el usuario tiene sus perfiles
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>El usuario debe estar logueado</li> <li>La educación continua es ofertada a un tipo de público distinto al perfil del usuario</li> </ul>
<b>Resultado esperado</b>	Inscripción fallida
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>Visualizar el panel de eventos activos</li> <li>Dar click en el nombre de educación continua activa para ampliar la información</li> <li>Visualizar mensaje de no poder realizar la inscripción</li> </ol>

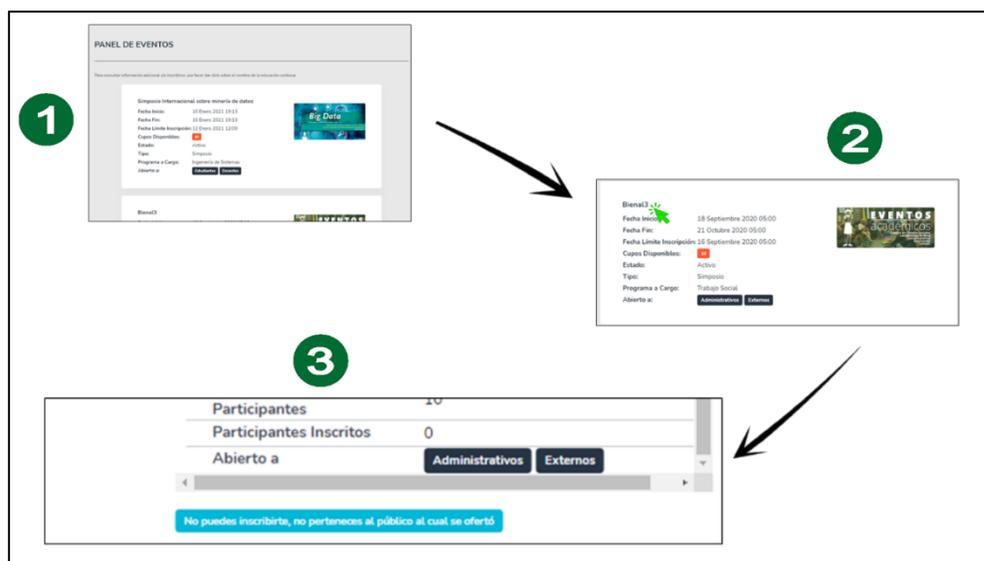


Figura 90 Resultado caso de prueba inscripción fallida a una educación continua sin cumplir requisitos

Tabla 66 Caso de prueba inscripción de un usuario a un curso o evento de educación continua con límite máximo cupos alcanzados

CASO DE PRUEBA	CP-013
	HU3
	19/04/2021
<b>Nombre</b>	Inscripción de un usuario a un curso o evento de educación continua con límite máximo alcanzado
<b>Descripción</b>	El usuario intenta inscribirse a un curso o evento de educación continua sin cupos disponibles
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>El usuario debe estar logueado</li> <li>El curso o evento de educación continua, si y solo tiene un límite de participantes, debe tener el cupo máximo de participantes inscritos</li> </ul>
<b>Resultado esperado</b>	Inscripción fallida
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>Visualizar el panel de eventos activos</li> <li>Dar click en el nombre de educación continua activa para ampliar la información</li> <li>Visualizar mensaje de no poder realizar la inscripción</li> </ol>

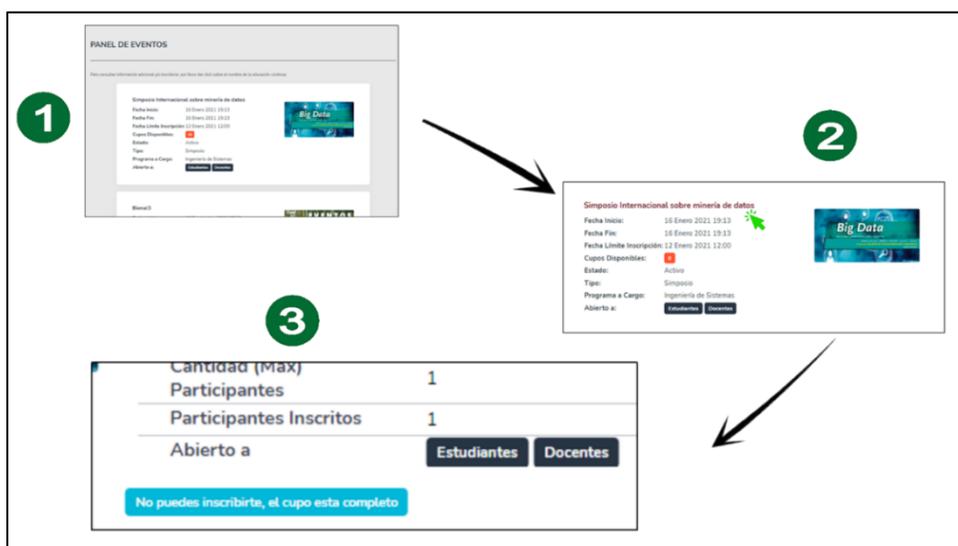


Figura 91 Resultado caso de prueba inscripción fallida por cupo máximo

Tabla 67 Caso de prueba inscripción de un usuario a un curso o educación continua fuera de fecha límite

<b>CASO DE PRUEBA</b>	<b>CP-014</b>
	<b>HU3</b>
	<b>19/04/2021</b>
<b>Nombre</b> Inscripción de un usuario a un curso o evento de educación continua fuera de fecha límite de inscripción	
<b>Descripción</b> El usuario intenta inscribirse a un curso o evento de educación continua disponible	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El curso o evento de educación continua debe tener una fecha límite de inscripción inferior a la actual</li> </ul>	
<b>Resultado esperado</b> Inscripción fallida	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Dar click en una educación continua activa</li> <li>2. Visualizar que la fecha límite de inscripción de la educación continua es inferior a la actual</li> <li>3. Visualizar mensaje de no poder realizar la inscripción</li> </ol>	

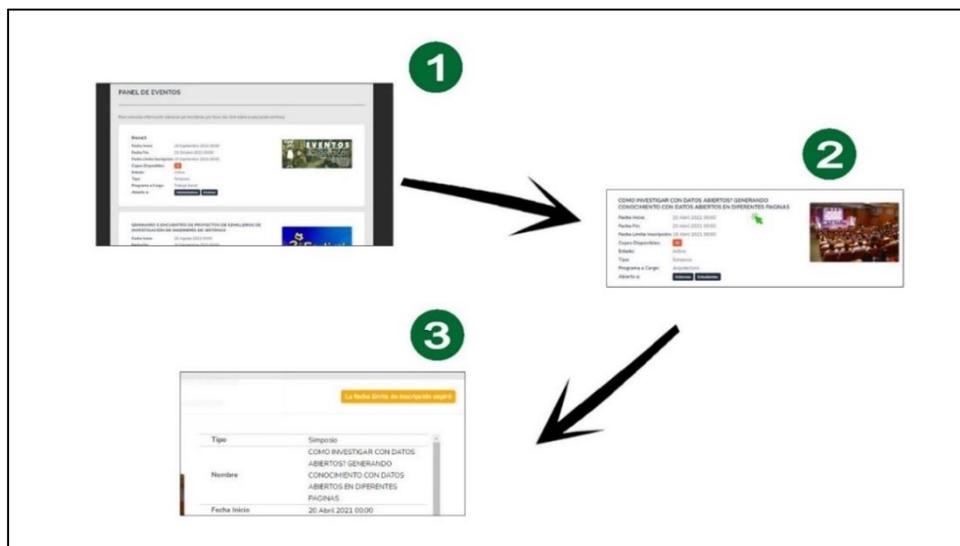


Figura 92 Resultado caso de prueba inscripción fallida por fecha límite de inscripción superada.

## ANEXO 2. ITERACIÓN 2 FASE DE DESARROLLO

### ANEXO 2-1. HU4 Generar reportes de SNIES

#### Planeación de tareas en tarjeta Kanban

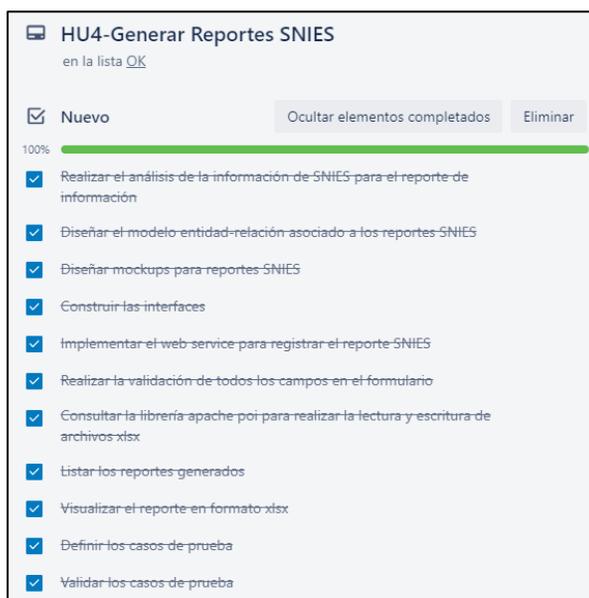


Figura 93 Tarjeta Kanban generar

#### Definición del modelo de datos

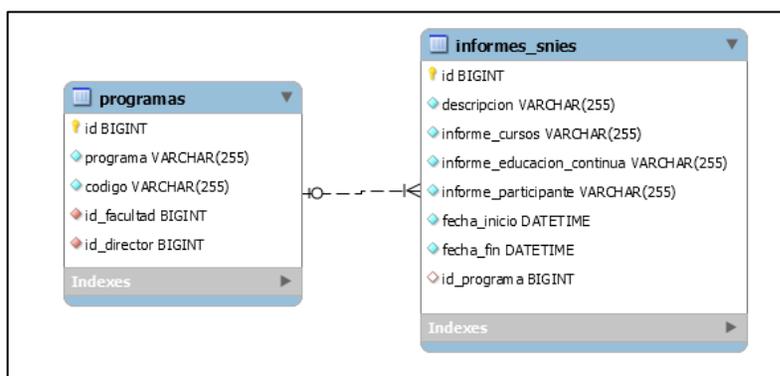


Figura 94 Modelo Entidad-Relación

## Diseño de mockups

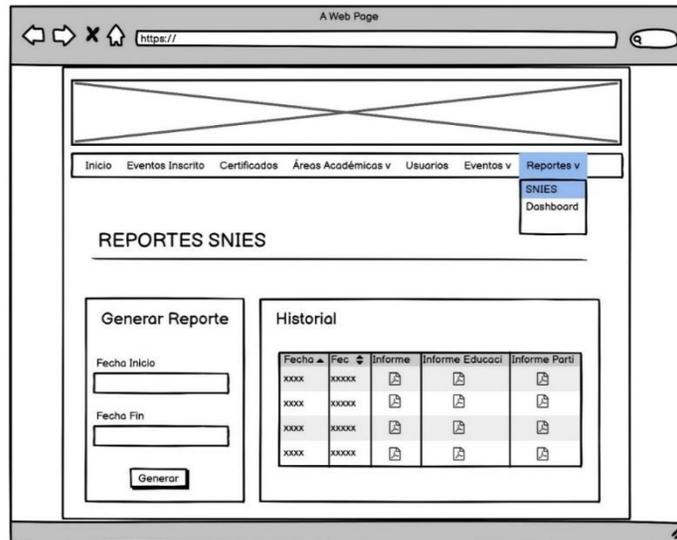


Figura 95 Mockup generar reportes SNIES.

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IInformeSniesService informeSniesService;

@InjectMocks
private ReportesSniesRestController reporteSniesController;

private static InformeSniesDto INFORME_SNIES_DTO;
private static String FECHA_INICIO;
private static String FECHA_FIN;
private static String DESCRIPCION;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    INFORME_SNIES_DTO = new InformeSniesDto();
    FECHA_INICIO = "2020-01-05";
    FECHA_FIN = "2020-06-05";
    DESCRIPCION = "Reporte Educación Continua Primer Semestre 2020";
    INFORME_SNIES_DTO.setFechaInicio(FECHA_INICIO);
    INFORME_SNIES_DTO.setFechaFin(FECHA_FIN);
    INFORME_SNIES_DTO.setDescripcion(DESCRIPCION);
}

@Test
public void generarReporteTest() {
    Mockito.doNothing().when(informeSniesService).generarReporteSNIES(
        FECHA_INICIO, FECHA_FIN, DESCRIPCION);
    final ResponseEntity? response = reporteSniesController.informeExcel(
        INFORME_SNIES_DTO);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 96 Test generar reportes SNIES.

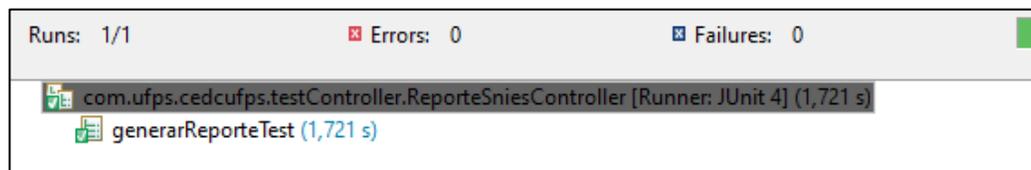


Figura 97 Resultado test generar reportes

- Pruebas de aceptación

Tabla 68 Caso de prueba generar reportes SNIES

<b>CASO DE PRUEBA</b>	<b>CP-015</b>								
	<b>HU4</b>								
	<b>19/04/2021</b>								
<b>Nombre</b> Generar reportes SNIES									
<b>Descripción</b> El usuario selecciona fecha de inicio y fecha fin para generar los reportes de SNIES									
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_SNIES</li> </ul>									
<b>Resultado esperado</b> Registro exitoso									
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Dar click en el menú Reportes, submenú SNIES</li> <li>2. Diligenciar el formulario correctamente para generar el reporte <table border="1" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;">Campo</th> <th style="text-align: center;">Valor</th> </tr> </thead> <tbody> <tr> <td>Fecha Inicio</td> <td>15/02/2020 08:00</td> </tr> <tr> <td>Fecha Fin</td> <td>15/06/2020 18:00</td> </tr> <tr> <td>Descripción</td> <td>Reporte Semestre 1 2020</td> </tr> </tbody> </table> </li> <li>3. Dar click en guardar</li> <li>4. Visualizar el mensaje de registro exitoso</li> </ol>		Campo	Valor	Fecha Inicio	15/02/2020 08:00	Fecha Fin	15/06/2020 18:00	Descripción	Reporte Semestre 1 2020
Campo	Valor								
Fecha Inicio	15/02/2020 08:00								
Fecha Fin	15/06/2020 18:00								
Descripción	Reporte Semestre 1 2020								

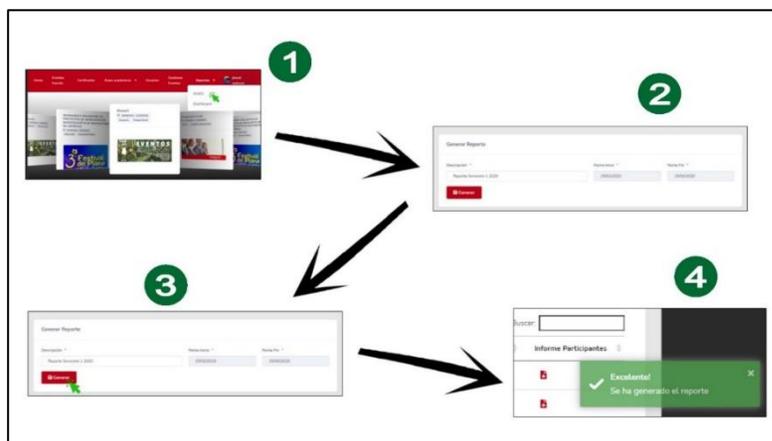


Figura 98 Resultado caso de prueba generar reportes SNIES.

Tabla 69 Caso de prueba visualizar listado de historial de informes SNIES generados

CASO DE PRUEBA	CP-016
	HU4
	19/04/2021
<b>Nombre</b> Visualizar listado de historial de informes SNIES generados	
<b>Descripción</b> El usuario visualiza el listado de los reportes SNIES (cursos, educación continua y participantes) generados en un rango de fecha en específico	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_SNIES</li> </ul>	
<b>Resultado esperado</b> Listado reportes SNIES	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Dar click en el menú Reportes, submenú SNIES</li> <li>2. Visualizar el listado del historial de reportes generado</li> </ol>	



Figura 99 Resultado caso de prueba visualizar reportes SNIES

Tabla 70 Caso de prueba descarga de informes SNIES generado

<b>CASO DE PRUEBA</b>	<b>CP-017</b>
	<b>HU4</b>
	<b>19/04/2021</b>
<b>Nombre</b> Descarga de informe SNIES generado	
<b>Descripción</b> El usuario descarga el informe de cursos, educación continua y/o participantes generados en formato .xlsx	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_SNIES</li> </ul>	
<b>Resultado esperado</b> Descarga de archivo	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de reportes SNIES</li> <li>2. Dar click en el botón  para descargar el archivo del reporte que desee</li> <li>3. Visualizar el archivo descargado</li> </ol>	

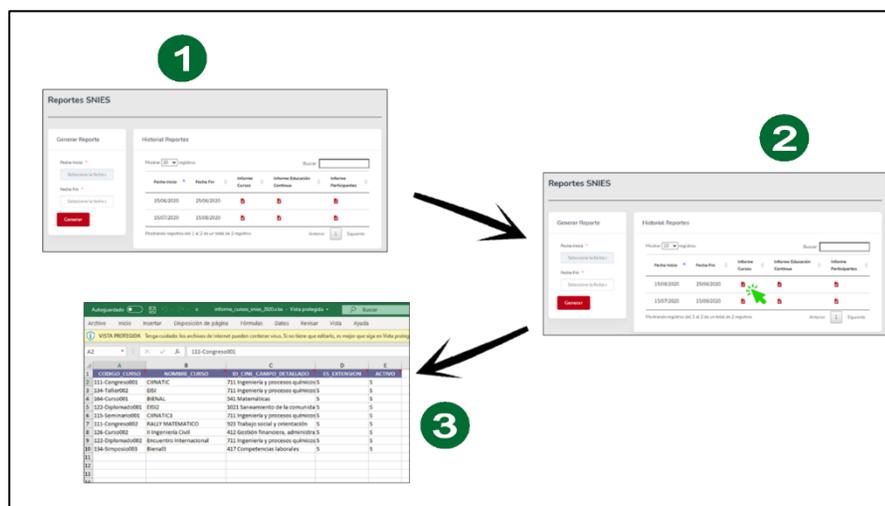


Figura 100 Resultado caso de prueba descarga de reporte SNIES

## ANEXO 2-2. HU13 Cancelar Inscripción

### Planeación de tareas en tarjeta Kanban

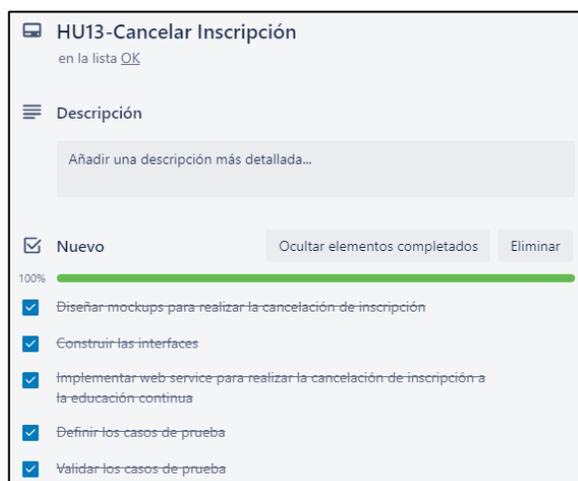


Figura 101 Tarjeta Kanban cancelar inscripción.

### Diseño de mockups

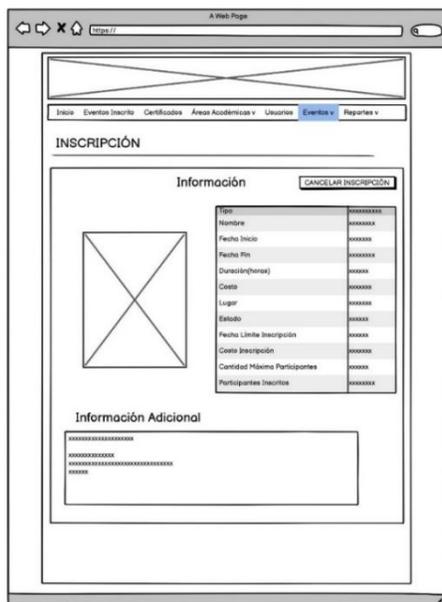


Figura 102 Mockup cancelar inscripción

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IParticipanteService participanteService;

@InjectMocks
private AsistenteRestController asistenteRestController;

private static String ID_EDU_CONTINUA;
private static String ID_TIPO_PERSONA;
private static ParticipanteDto PARTICIPANTE_DTO;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
    ID_EDU_CONTINUA="1";
    ID_TIPO_PERSONA="1";
    PARTICIPANTE_DTO=new ParticipanteDto();
    PARTICIPANTE_DTO.setId(1L);
    PARTICIPANTE_DTO.setIdEducacionContinua(Long.parseLong(ID_EDU_CONTINUA));
    PARTICIPANTE_DTO.setIdTipoPersona(Long.parseLong(ID_TIPO_PERSONA));
    PARTICIPANTE_DTO.setIdPersona(1L);
}

```

Figura 104 Instanciación variables test cancelar

```

@Test
public void cancelarInscripcionTest() {
    Mockito.doNothing().when(participanteService).cancelarInscripcion
(Long.parseLong(ID_EDU_CONTINUA));
    final ResponseEntity<?> response = asistenteRestController.cancelarInscripcion
(Long.parseLong(ID_EDU_CONTINUA));
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 103 Test cancelar inscripción.

- Pruebas de aceptación

Tabla 71 Caso de prueba cancelar inscripción de curso o evento de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-031</b>
	<b>HU13</b>
	<b>19/04/2021</b>

<b>Nombre</b> Cancelar inscripción de curso o evento de educación continua
<b>Descripción</b> El usuario cancela su inscripción a un curso o evento de educación continua.
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe estar inscrito al curso o evento de educación continua para cancelar la inscripción</li> </ul>
<b>Resultado esperado</b> Inscripción cancelada
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Dar click en una educación continua activa a la cuál ya se encuentre inscrito</li> <li>2. Dar click en el botón “Cancelar Inscripción”</li> <li>3. Visualizar mensaje de cancelación inscripción exitosamente</li> </ol>



Figura 105 Resultado caso de prueba cancelar inscripción a una educación

## ANEXO 2-3. HU17 Login del sistema

## Planeación de tareas en tarjeta Kanban

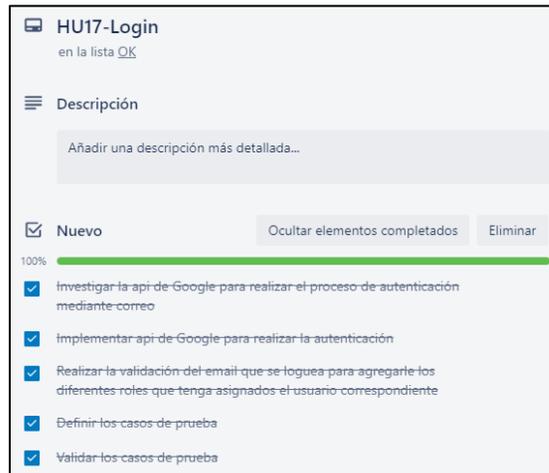


Figura 106 Tarjeta Kanban login

## Diseño de mockups

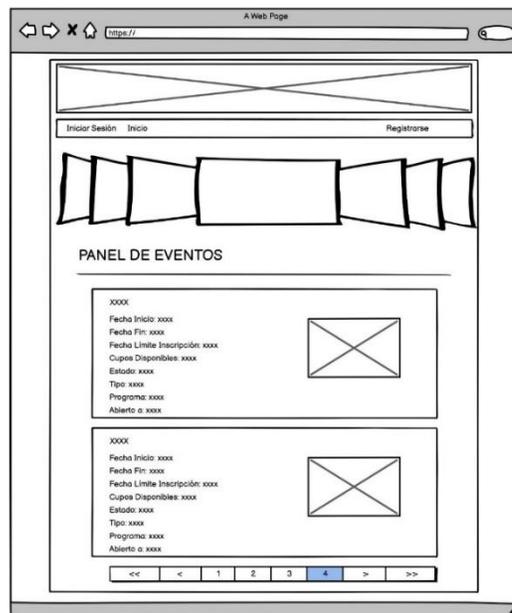


Figura 107 Mockup inicio sistema

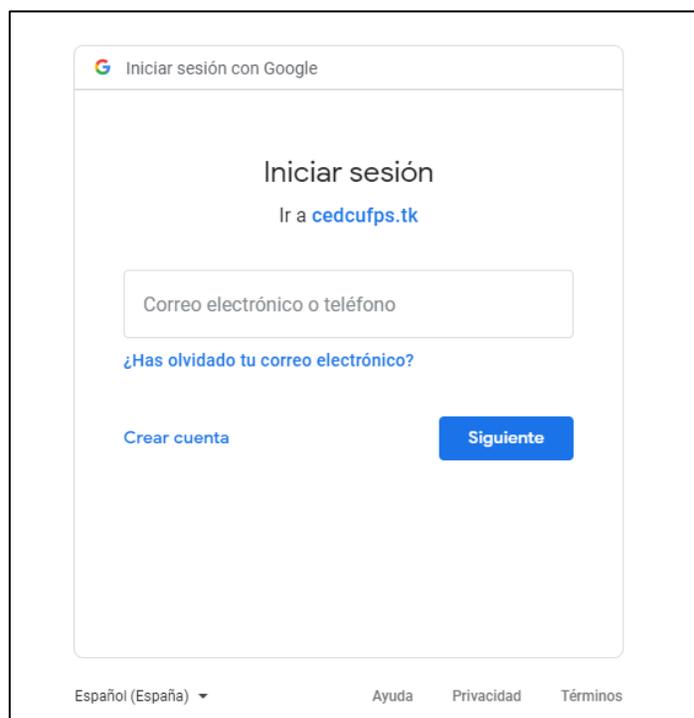


Figura 108 Interfaz Login del sistema.

## Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 72 Caso de prueba login del sistema con un usuario existente

<b>CASO DE PRUEBA</b>	<b>CP-048</b>
	<b>HU17</b>
	<b>19/04/2021</b>
<b>Nombre</b> Login del sistema de información con usuario existente	
<b>Descripción</b> El usuario se loguea mediante autenticación Google.	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar registrado en el sistema y tener correo de Google</li> </ul>	
<b>Resultado esperado</b> Ingreso exitoso	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Dar click en el botón “Iniciar Sesión”</li> <li>2. Ingresar las credenciales del correo registrado</li> <li>3. Ingreso exitoso</li> </ol>	

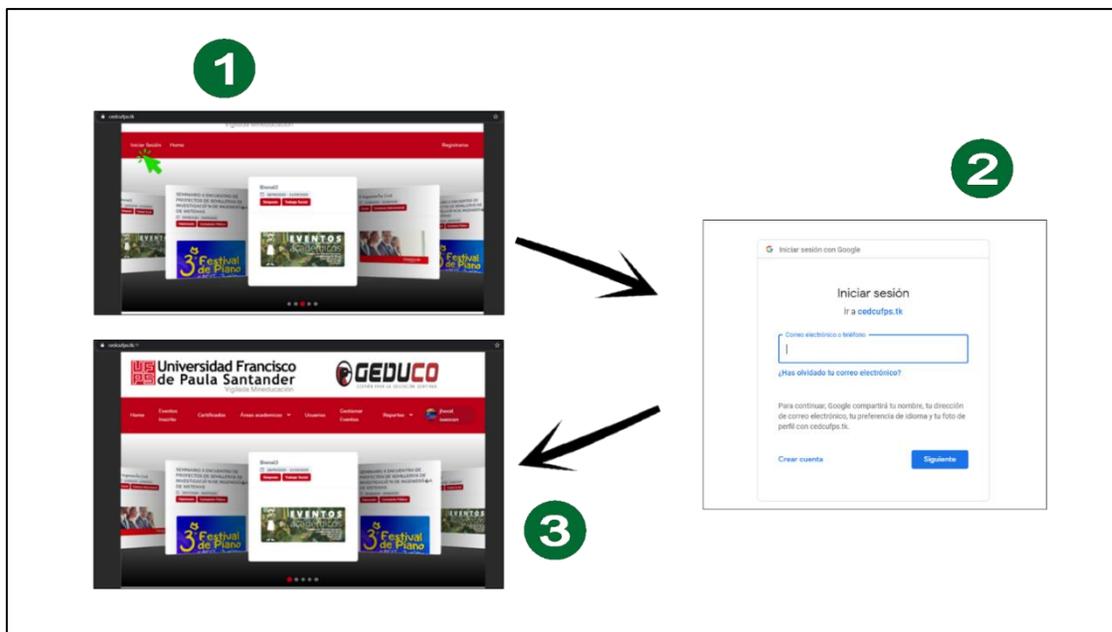


Figura 109 Resultado caso de prueba login del sistema con usuario existente.

Tabla 73 Login del sistema con usuario nuevo

<b>CASO DE PRUEBA</b>	<b>CP-049</b>
	<b>HU17</b>
	<b>19/04/2021</b>
<b>Nombre</b>	Login del sistema de información con usuario nuevo
<b>Descripción</b>	El usuario se loguea mediante autenticación Google.
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>El usuario no debe estar registrado en el sistema</li> </ul>
<b>Resultado esperado</b>	Ingreso exitoso y creación de nuevo usuario
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>1. Dar click en el botón “Iniciar Sesión”</li> <li>2. Ingresar las credenciales del correo registrado</li> <li>3. Diligenciar los campos de registro de nuevo usuario</li> </ol>

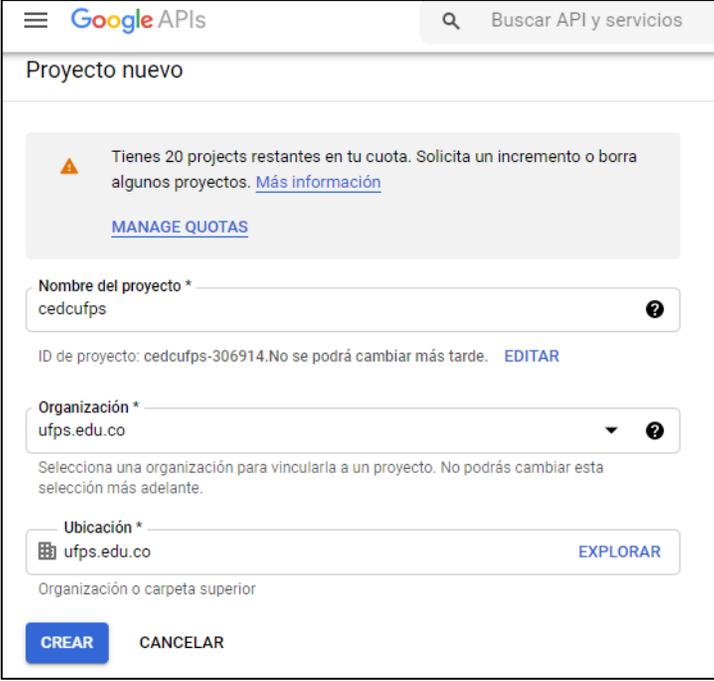


## ANEXO 2-4. Configuración API Login Google

- 1) Loguearse en <http://console.developers.google.com/> para realizar las configuraciones
- 2) Crear un proyecto nuevo



- 3) Completar el formulario de registro



The screenshot shows the 'Proyecto nuevo' (New Project) registration form in the Google Cloud Platform console. The form includes the following fields and buttons:

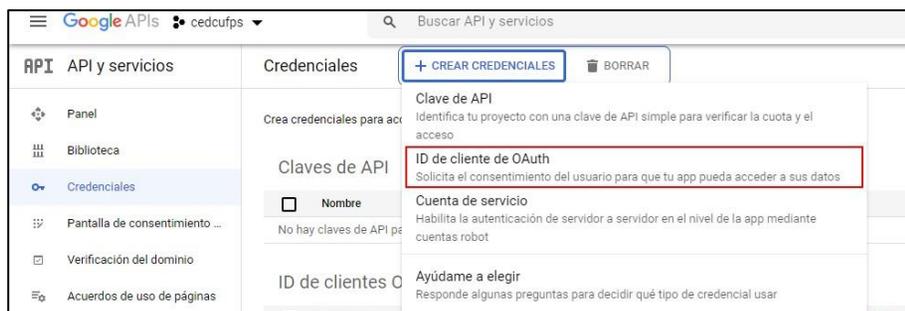
- Nombre del proyecto \***: Input field containing 'cedcufps'.
- ID de proyecto**: Displayed as 'cedcufps-306914.No se podrá cambiar más tarde. EDITAR'.
- Organización \***: Dropdown menu containing 'ufps.edu.co'.
- Ubicación \***: Input field containing 'ufps.edu.co' with an 'EXPLORAR' button next to it.
- Buttons**: 'CREAR' (Create) and 'CANCELAR' (Cancel) buttons at the bottom.

A warning message at the top states: 'Tienes 20 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. Más información' with a 'MANAGE QUOTAS' link.

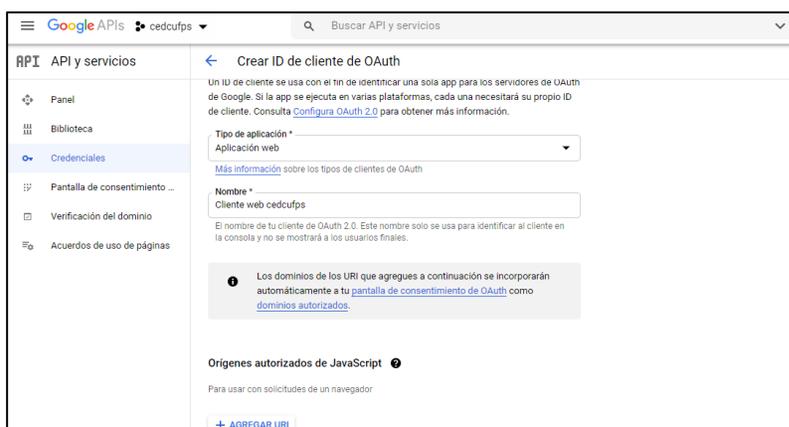
4) Dar click en la opción “credenciales” del menú lateral



5) Dar click en crear credenciales, opción “ID de cliente de OAuth”



6) Diligenciar el formulario para generar la credencial del login



- 7) Definir las URI de direccionamiento; la primera, `http://localhost:8080/login/oauth2/code/google`, es para la fase de desarrollo y la segunda, `https://cedcufps.tk/login/oauth2/code/google`, para fase de producción. Es importante resaltar que el dominio `cedcufps.tk` se generó de manera gratuita a través de `freenom.com`. Posterior a esto, dar click en guardar y se generan las keys necesarias para nuestro entorno de desarrollo.



URI de redireccionamiento autorizados ⓘ

Para usar con solicitudes de un servidor web

URI

`http://localhost:8080/login/oauth2/code/google`

`https://cedcufps.tk/login/oauth2/code/google`

+ AGREGAR URI

GUARDAR CANCELAR

- 8) Una vez creada, se genera el ID de cliente y el cliente secreto como lo indica la siguiente figura

<b>ID de cliente</b>	226613155821-ri427lmire8qq5icol9s20srnuvh2vqg.apps.googleusercontent.com
<b>Secreto del cliente</b>	bTJqrfgRmLIwmGFvV_BnFTFd
<b>Fecha de creación</b>	25 de febrero de 2020 a las 10:37:11 GMT-5

- 9) Dichas keys generadas anteriormente, se ingresan en las propiedades del proyecto de spring boot de la siguiente manera

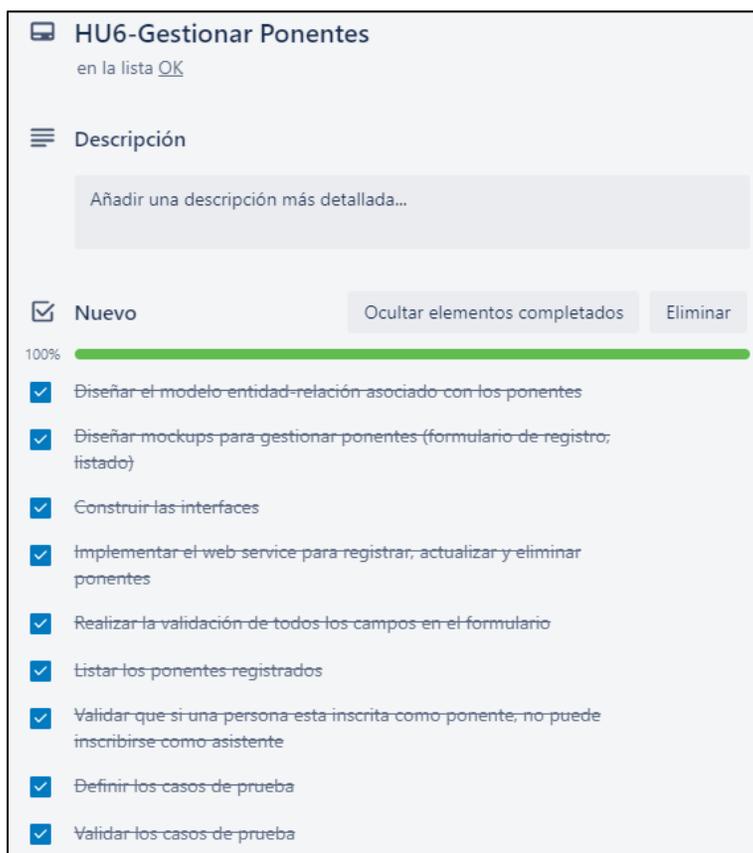
```
11 spring.security.oauth2.client.registration.google.client-id=226613155821-ri427lmire8qq5icol9s20srnuvh2vqg.apps.googleusercontent.com
12 spring.security.oauth2.client.registration.google.client-secret=bTJqrfgRmLIwmGFvV_BnFTFd
```

Una vez configurado el proyecto, se realiza la codificación y las pruebas pertinentes para validar el funcionamiento correcto del login a través de Google

## ANEXO 3. ITERACIÓN 3 FASE DE DESARROLLO

## ANEXO 3-1. HU6 Gestionar ponentes de la educación continua

## Planeación de tareas en tarjeta Kanban



**HU6-Gestionar Ponentes**  
en la lista [OK](#)

Descripción

Añadir una descripción más detallada...

**Nuevo** Ocultar elementos completados Eliminar

100%

- Diseñar el modelo entidad-relación asociado con los ponentes
- Diseñar mockups para gestionar ponentes (formulario de registro, listado)
- Construir las interfaces
- Implementar el web service para registrar, actualizar y eliminar ponentes
- Realizar la validación de todos los campos en el formulario
- Listar los ponentes registrados
- Validar que si una persona esta inscrita como ponente, no puede inscribirse como asistente
- Definir los casos de prueba
- Validar los casos de prueba

Figura 111 Tarjeta Kanban gestionar ponentes.

## Definición del modelo de datos

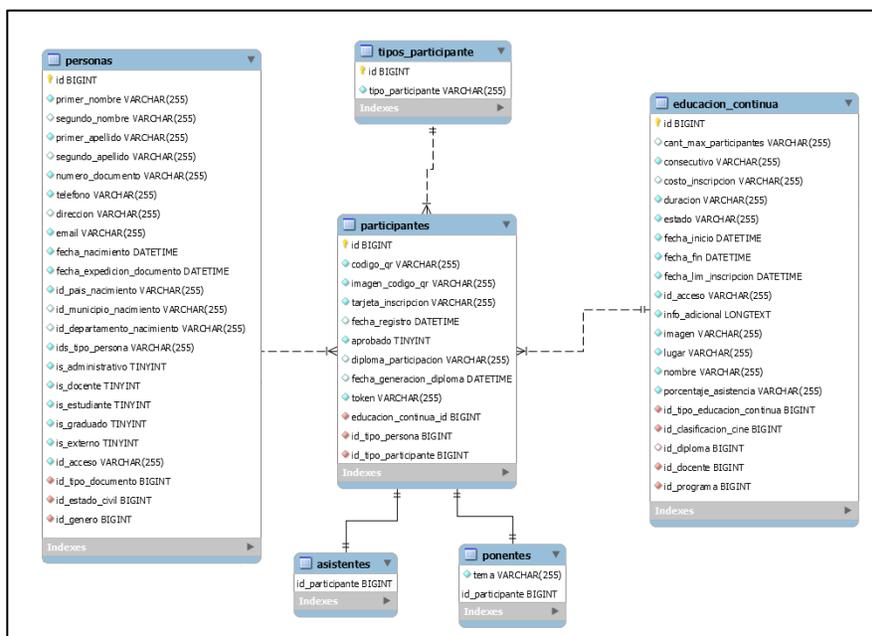


Figura 112 Modelo Entidad-Relación ponentes

## Diseño de mockups



Figura 113 Mockup listar

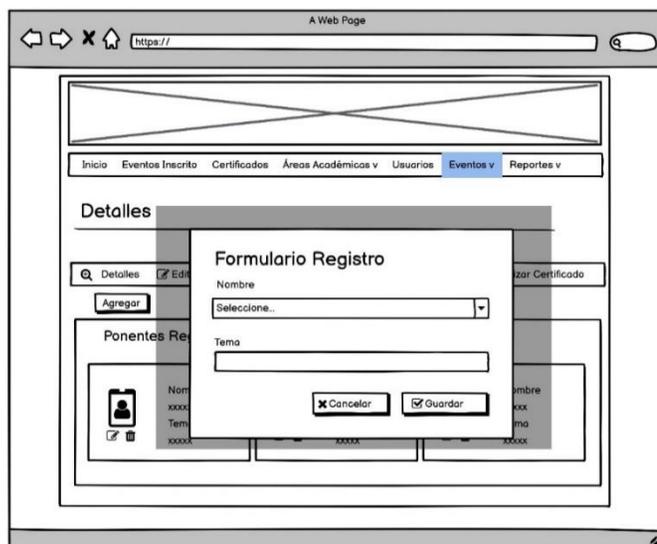


Figura 114 Mockup registro ponentes

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IParticipanteService participanteService;

@Mock
private IPersonaService personaService;

@InjectMocks
private PonenteRestController ponenteController;

private static Ponente PONENTE;
private static PonenteDto PONENTE_DTO;
private static PersonaDto PERSONA_DTO;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);

    PONENTE_DTO= new PonenteDto();
    ParticipanteDto participante= new ParticipanteDto();
    participante.setIdEducacionContinua(1L);
    participante.setIdPersona(1L);
    participante.setIdTipoParticipante(2L);
    PONENTE_DTO.setParticipante(participante);
    PONENTE_DTO.setTema("Fundamentos de Programación");

    PERSONA_DTO=new PersonaDto();
    PERSONA_DTO.setId(1L);
    PERSONA_DTO.setDocumento("54677845");
    PERSONA_DTO.setNombre("Francisco Perez");

    PONENTE= new Ponente();
    PONENTE.setTema("Ciclos en Java");
}

```

Figura 115 Instanciación variables test gestionar

```

@Test
public void savePonente() {
    final ResponseEntity<?> response = ponenteController.guardarPonenteRest(PONENTE);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void searchPersona() {
    Mockito.when(personaService.findOne(1L)).thenReturn(PERSONA_DTO);
    final ResponseEntity<PersonaDto> response = ponenteController.findPersona("1");
    assertNotNull(response.getBody());
    assertEquals(response.getBody().getDocumento(), PERSONA_DTO.getDocumento());
}

@Test
public void deletePonente() {
    final ResponseEntity<?> response = ponenteController.deletePonente("1");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void buscarPonente() {
    Mockito.when(participanteService.findPonente(1L)).thenReturn(PONENTE_DTO);
    final ResponseEntity<PonenteDto> response = ponenteController.buscarPonente(1L);
    assertNotNull(response.getBody());
    assertEquals(response.getBody().getTema(), PONENTE_DTO.getTema());
}

```

Figura 117 Test gestionar ponentes

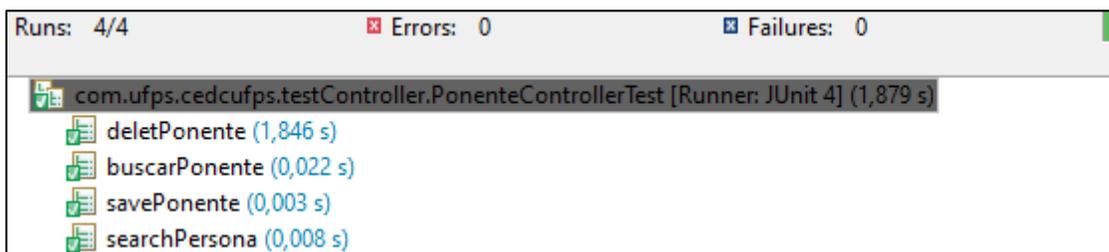


Figura 116 Resultados test gestionar ponentes

- Pruebas de aceptación

Tabla 74 Caso de prueba registro exitoso de un ponente

CASO DE PRUEBA	CP-019
	HU6
	19/04/2021
<b>Nombre</b> Registro exitoso de un ponente	
<b>Descripción</b> El usuario registra un ponente a un evento o curso de educación continua diligenciando el formulario correctamente	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> </ul>	

<ul style="list-style-type: none"> <li>El usuario debe tener rol: ROLE_MANAECCU</li> </ul>																
<b>Resultado esperado</b> Registro exitoso																
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>Ubicarse en la interfaz de gestionar eventos</li> <li>Dar click en el botón  para visualizar detalles de la educación continua</li> <li>Dar click en la opción “ponentes”</li> <li>Realizar el filtro de búsqueda para asignar una persona de las que están registradas a ponentes</li> </ol> <table border="1" data-bbox="407 611 971 722"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Tipo de búsqueda</td> <td>Nombre</td> </tr> <tr> <td>Valor</td> <td>Pedro</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>Seleccionar la persona para agregar al ponente al dar click en el botón “agregar ponente”</li> <li>Diligenciar el formulario de registro de ponente</li> </ol> <table border="1" data-bbox="407 867 971 1125"> <thead> <tr> <th>Campo</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>Tipo de documento</td> <td>CC</td> </tr> <tr> <td>Documento</td> <td>1561</td> </tr> <tr> <td>Nombre</td> <td>Pedro Jose Llanos Ortega</td> </tr> <tr> <td>Tema</td> <td>Ciclos for, while, do while en Java</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>Dar click en guardar</li> <li>Visualizar el mensaje de registro exitoso</li> </ol>	Campo	Valor	Tipo de búsqueda	Nombre	Valor	Pedro	Campo	Valor	Tipo de documento	CC	Documento	1561	Nombre	Pedro Jose Llanos Ortega	Tema	Ciclos for, while, do while en Java
Campo	Valor															
Tipo de búsqueda	Nombre															
Valor	Pedro															
Campo	Valor															
Tipo de documento	CC															
Documento	1561															
Nombre	Pedro Jose Llanos Ortega															
Tema	Ciclos for, while, do while en Java															

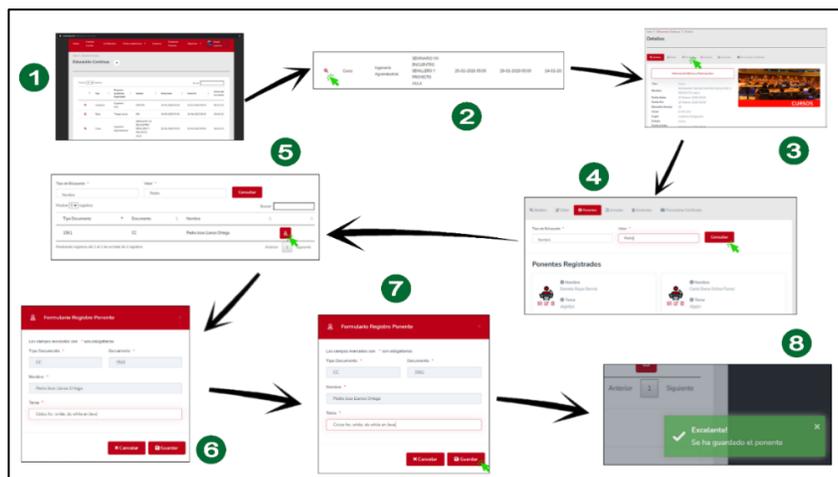


Figura 118 Caso de prueba registro exitoso de un ponente.

Tabla 75 Caso de prueba visualizar listado de ponentes registrados

CASO DE PRUEBA	CP-020
	HU6
	19/04/2021
<b>Nombre</b> Visualizar listado de ponentes registrados	
<b>Descripción</b> El usuario visualiza los ponentes registrados en evento o curso de educación continua	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>	
<b>Resultado esperado</b> Listado ponentes	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “ponentes”</li> <li>4. Visualizar el listado de ponentes registrado</li> </ol>	

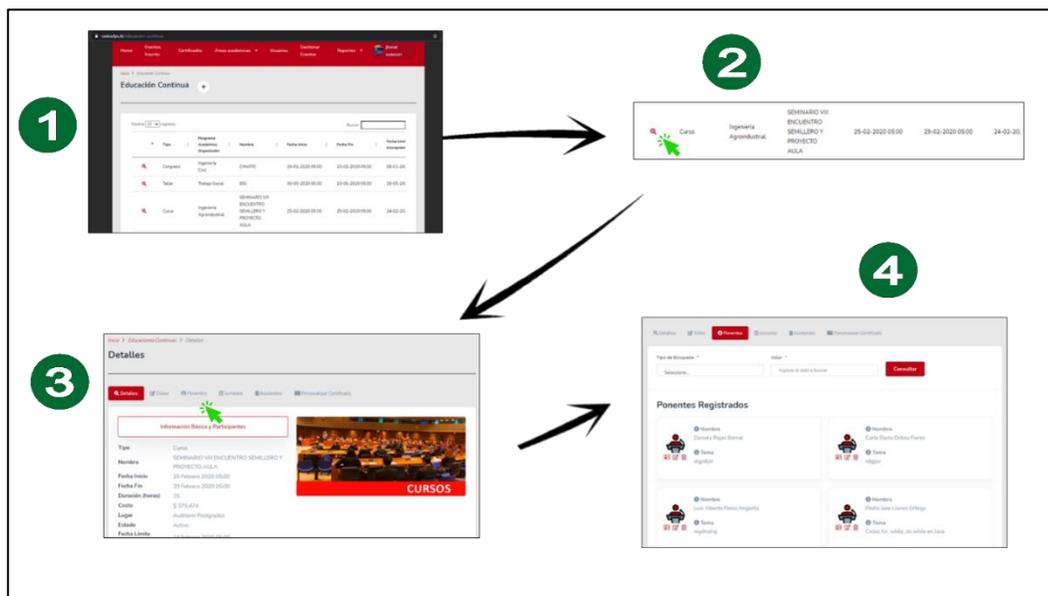


Figura 119 Resultado caso de prueba visualizar listado ponentes

Tabla 76 Caso de prueba actualización exitosa de un ponente

CASO DE PRUEBA	CP-021						
	HU6						
	19/04/2021						
<b>Nombre</b> Actualización exitosa de un ponente							
<b>Descripción</b> El usuario actualiza la información de un ponente a un evento o curso de educación continua diligenciando el formulario correctamente							
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>							
<b>Resultado esperado</b> Actualización exitosa							
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “ponentes”</li> <li>4. Visualizar el listado de ponentes registrado</li> <li>5. Dar click en  sobre el ponente que se va a actualizar</li> <li>6. Editar el tema del ponente</li> </ol> <table border="1" data-bbox="407 1083 1256 1192"> <thead> <tr> <th>Campo</th> <th>Valor Antigo</th> <th>Valor Nuevo</th> </tr> </thead> <tbody> <tr> <td>Tema</td> <td>Ciclos for, while, do while en Java</td> <td>Estructuras de datos</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>7. Dar click en guardar</li> <li>8. Visualizar el mensaje de actualización exitosa</li> </ol>	Campo	Valor Antigo	Valor Nuevo	Tema	Ciclos for, while, do while en Java	Estructuras de datos	
Campo	Valor Antigo	Valor Nuevo					
Tema	Ciclos for, while, do while en Java	Estructuras de datos					

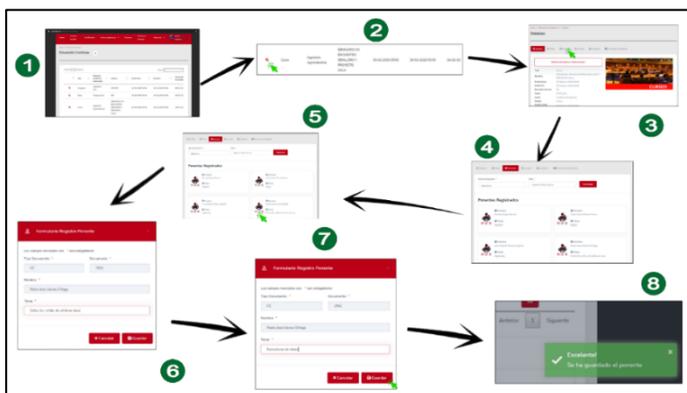


Figura 120 Resultado caso de prueba actualización exitosa de

## ANEXO 3-2. HU7 Gestionar jornadas de la educación continua

## Planeación de tareas en tarjeta Kanban

The image shows a Kanban card titled "HU7-Gestionar Jornadas" with the subtitle "en la lista OK". It features a "Descripción" section with a placeholder "Añadir una descripción más detallada...". Below this is a "Nuevo" section with a 100% progress bar and two buttons: "Ocultar elementos completados" and "Eliminar". A list of eight tasks follows, each with a checked checkbox:

- ✓ Diseñar el modelo entidad-relación asociado con las jornadas
- ✓ Diseñar mockups para gestionar jornadas (formulario de registro, listado)
- ✓ Construir las interfaces
- ✓ Implementar el web service para registrar, actualizar y eliminar jornadas
- ✓ Realizar la validación de todos los campos en el formulario
- ✓ Listar las jornadas registradas
- ✓ Definir los casos de prueba
- ✓ Validar los casos de prueba

Figura 121 Tarjeta Kanban gestionar jornadas

## Definición del modelo de datos

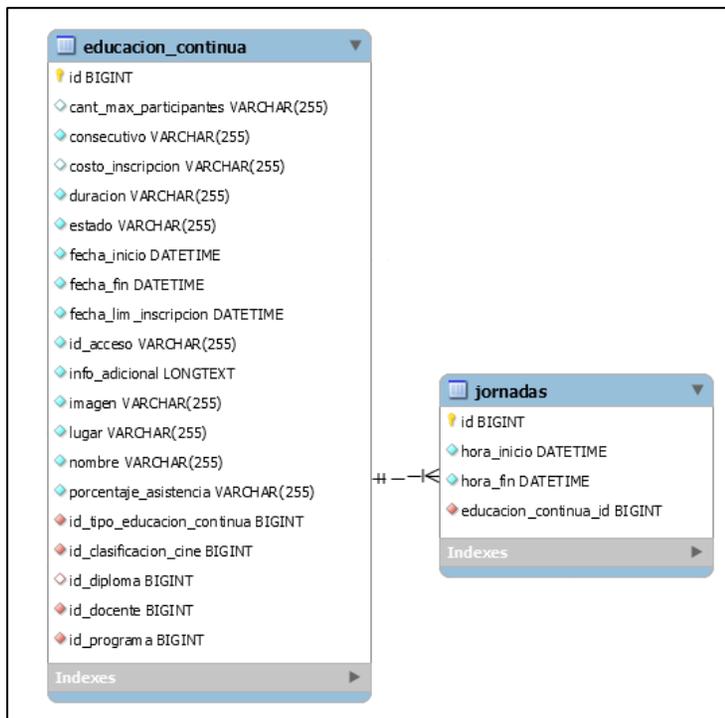


Figura 122 Modelo Entidad-Relación jornadas

## Diseño de mockups

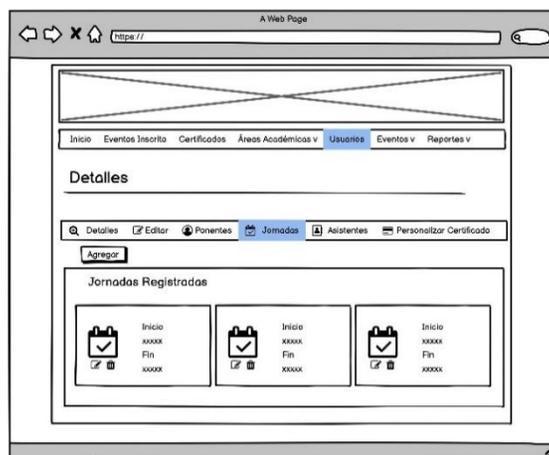


Figura 123 Mockup listar jornadas

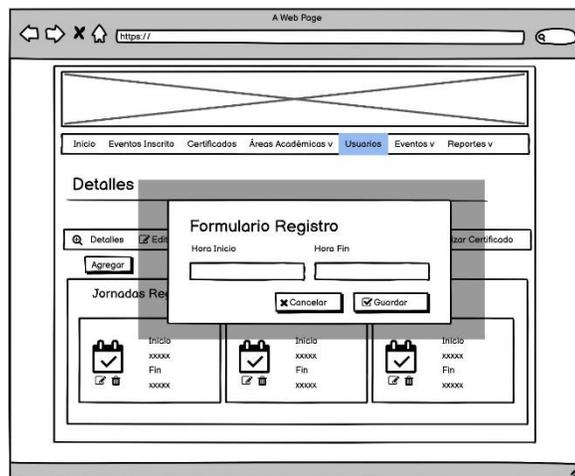


Figura 124 Mockup registro jornada.

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IJornadaService jornadaService;

@InjectMocks
private JornadaRestController jornadaController;

private static Jornada JORNADA1;
private static JornadaAppDto JORNADA_APP_DTO;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);

    JORNADA1= new Jornada();
    JORNADA1.setHoraInicio(new Date());
    JORNADA1.setHoraFin(new Date());

    JORNADA_APP_DTO= new JornadaAppDto();
    JORNADA_APP_DTO.setId(1L);
    JORNADA_APP_DTO.setHoraInicio(new Date());
    JORNADA_APP_DTO.setHoraFin(new Date());
}

```

Figura 125 Instanciación variables test

```

@Test
public void saveJornada() {
    final ResponseEntity<?> response = jornadaController.guardarJornada(JORNADA1);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void searchJornada() {
    Mockito.when(jornadaService.findOne(1L)).thenReturn(JORNADA_APP_DTO);
    final ResponseEntity<JornadaAppDto> response = jornadaController.buscarJornada(1L);
    assertNotNull(response.getBody());
    assertEquals(response.getBody().getHoraFin(), JORNADA_APP_DTO.getHoraFin());
}

@Test
public void deleteJornada() {
    final ResponseEntity<?> response = jornadaController.deleteJornada("1");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 127 Test gestionar jornadas

```

com.ufps.cedcufps.testController.JornadaControllerTest [Runner: JUnit 4] (1,792 s)
  deleteJornada (1,758 s)
  saveJornada (0,004 s)
  searchJornada (0,030 s)

```

Figura 126 Resultados test gestionar jornadas.

- Pruebas de aceptación

Tabla 77 Caso de prueba registro exitoso de una jornada

CASO DE PRUEBA	CP-022
	HU7
	19/04/2021
<b>Nombre</b> Registro exitoso de una jornada	
<b>Descripción</b> El usuario registra una jornada a un evento o curso de educación continua diligenciando el formulario correctamente	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>	
<b>Resultado esperado</b>	

Registro exitoso	
<b>Desarrollo</b>	
<ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “Jornadas”</li> <li>4. Dar click en el botón agregar para desplegar el formulario</li> <li>5. Diligenciar el formulario correctamente</li> </ol>	
<b>Campo</b>	<b>Valor</b>
Hora Inicio	11/12/2020 08:00
Hora Fin	11/12/2020 12:00
<ol style="list-style-type: none"> <li>6. Dar click en guardar</li> <li>7. Visualizar mensaje de registro exitoso</li> </ol>	

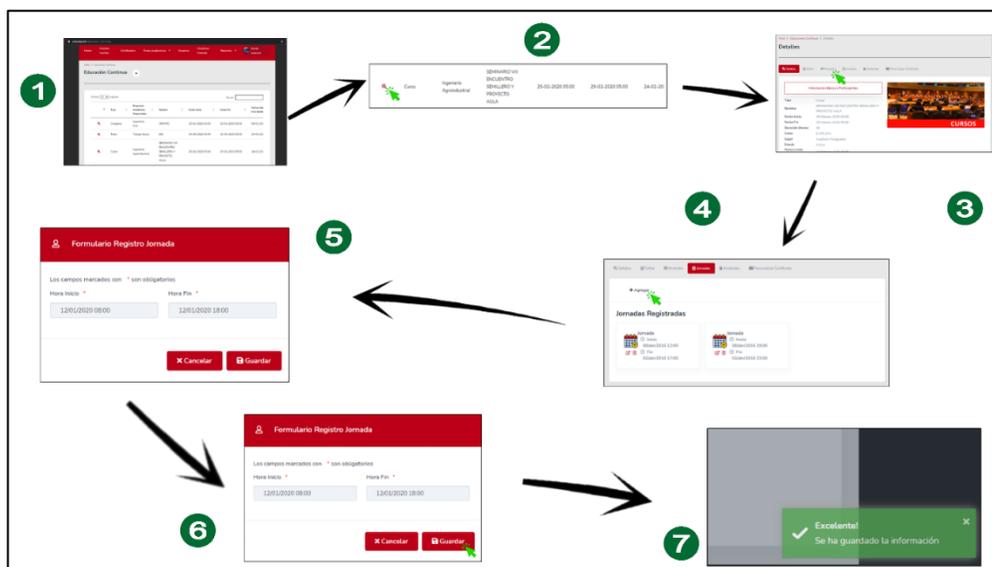


Figura 128 Resultado caso de prueba registro exitoso de

Tabla 78 Caso de prueba visualizar listado de jornadas registradas

<b>CASO DE PRUEBA</b>	<b>CP-023</b>
	<b>HU7</b>
	<b>19/04/2021</b>

<b>Nombre</b> Visualizar listado de jornadas registradas
<b>Descripción</b> El usuario visualiza las jornadas registradas en evento o curso de educación continua
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>
<b>Resultado esperado</b> Listado jornadas
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “Jornadas”</li> <li>4. Visualizar listado de jornadas registradas</li> </ol>

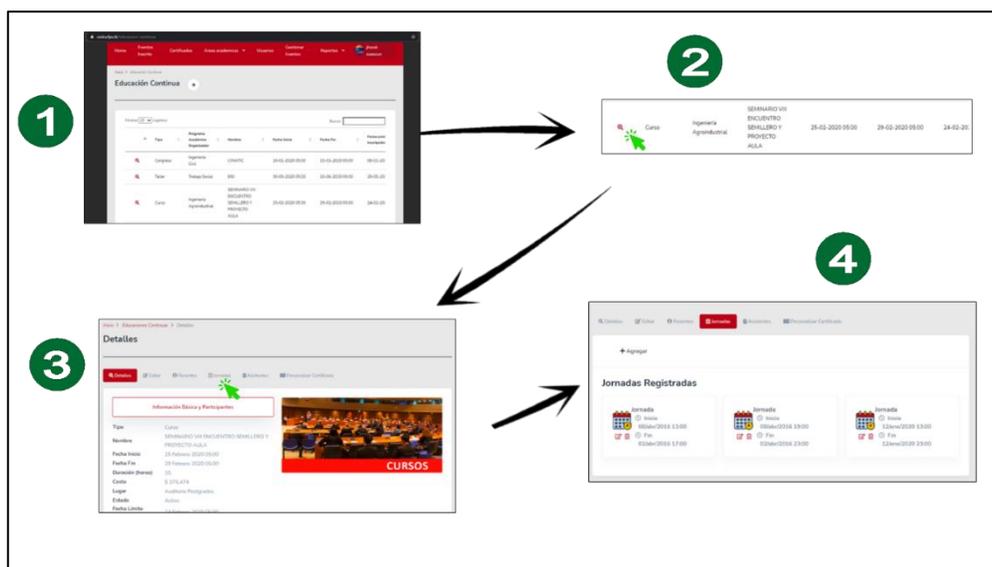


Figura 129 Resultado caso de prueba visualización de una

Tabla 79 Caso de prueba actualización exitosa de una jornada

<b>CASO DE PRUEBA</b>	<b>CP-024</b>
	<b>HU7</b>
	<b>19/04/2021</b>
<b>Nombre</b> Actualización exitosa de una jornada	

<p><b>Descripción</b></p> <p>El usuario actualiza la información de una jornada a un evento o curso de educación continua diligenciando el formulario correctamente</p>									
<p><b>Prerrequisitos</b></p> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>									
<p><b>Resultado esperado</b></p> <p>Actualización exitosa</p>									
<p><b>Desarrollo</b></p> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “Jornadas”</li> <li>4. Dar click en el icono  sobre la jornada que se va a actualizar</li> <li>5. Editar la jornada</li> </ol> <table border="1" data-bbox="406 798 1258 913"> <thead> <tr> <th>Campo</th> <th>Valor Antigo</th> <th>Valor Nuevo</th> </tr> </thead> <tbody> <tr> <td>Hora Inicio</td> <td>11/12/2020 08:00</td> <td>11/12/2020 14:00</td> </tr> <tr> <td>Hora Fin</td> <td>11/12/2020 12:00</td> <td>11/12/2020 18:00</td> </tr> </tbody> </table> <ol style="list-style-type: none"> <li>6. Dar click en guardar</li> <li>7. Visualizar el mensaje de actualización exitosa</li> </ol>	Campo	Valor Antigo	Valor Nuevo	Hora Inicio	11/12/2020 08:00	11/12/2020 14:00	Hora Fin	11/12/2020 12:00	11/12/2020 18:00
Campo	Valor Antigo	Valor Nuevo							
Hora Inicio	11/12/2020 08:00	11/12/2020 14:00							
Hora Fin	11/12/2020 12:00	11/12/2020 18:00							

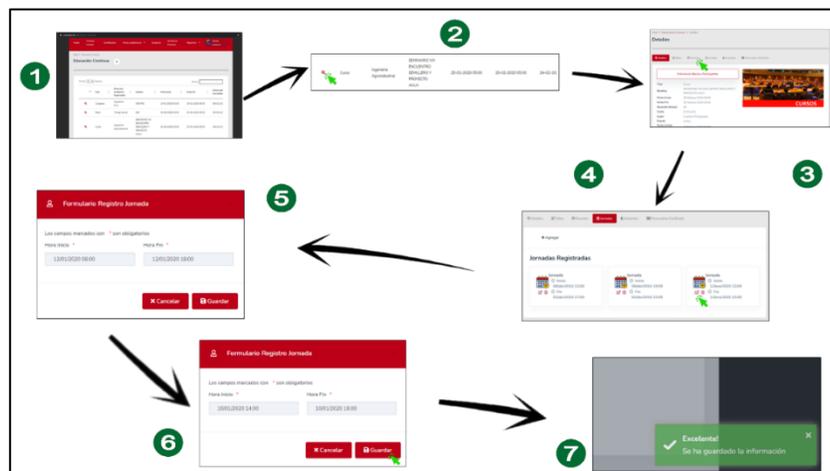


Figura 130 Resultado caso de prueba actualización de una

## ANEXO 3-3. HU11 Generar listado de asistentes

### Planeación de tareas en tarjeta Kanban

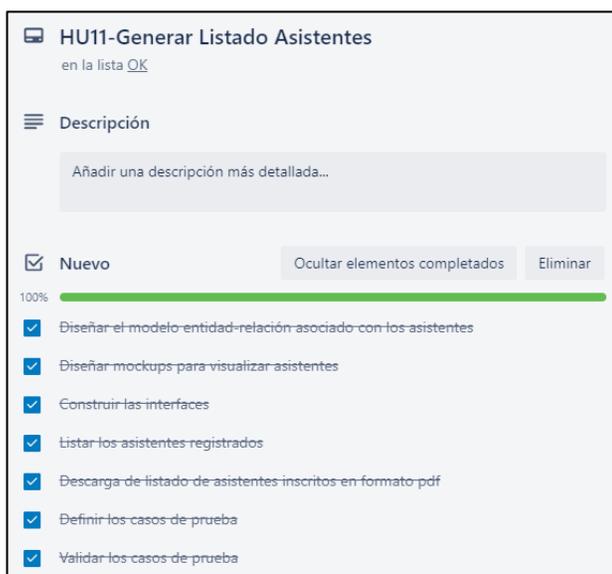


Figura 131 Tarjeta Kanban generar listado

### Definición del modelo de datos

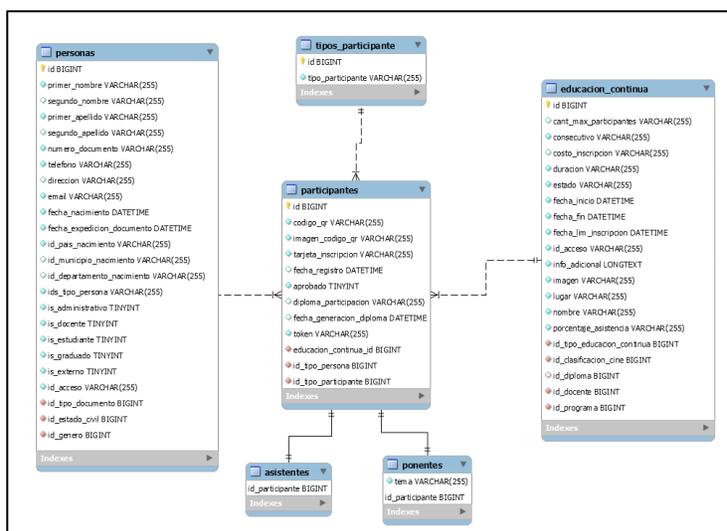


Figura 132 Modelo Entidad-Relación

## Diseño de mockups

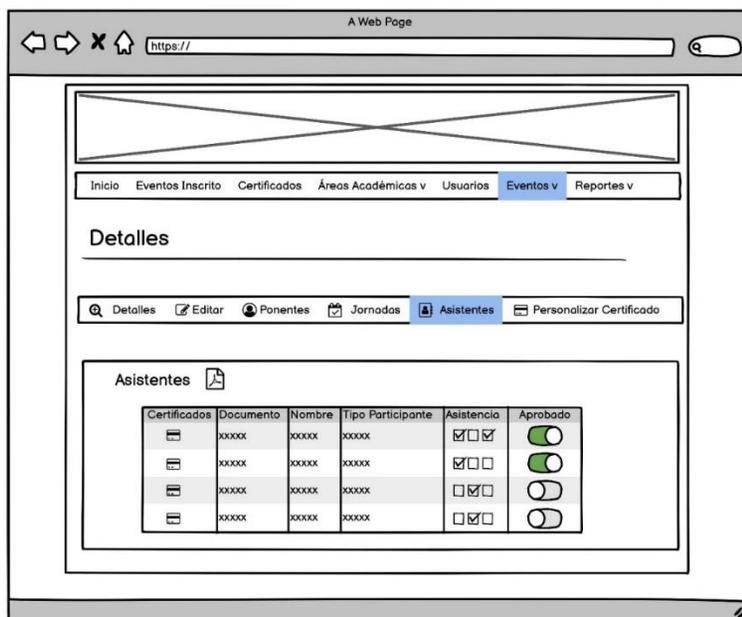


Figura 133 Mockup listado participantes.

## Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 80 Caso de prueba visualizar listado asistentes

<b>CASO DE PRUEBA</b>	<b>CP-028</b>
	<b>HU11</b>
	<b>19/04/2021</b>
<b>Nombre</b> Descargar listado de asistentes	
<b>Descripción</b> El usuario descarga el listado de participantes de cualquier curso o evento de educación continua.	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>	
<b>Resultado esperado</b> Descarga exitosa	
<b>Desarrollo</b>	

1. Ubicarse en la interfaz de gestionar eventos
2. Dar click en el botón  para visualizar detalles de la educación continua
3. Dar click en la opción “Asistentes”
4. Visualizar listado de asistentes inscritos en la educación continua
5. Dar click en el botón  para descargar el listado de asistentes inscritos
6. Visualizar el pdf generado de los asistentes inscritos

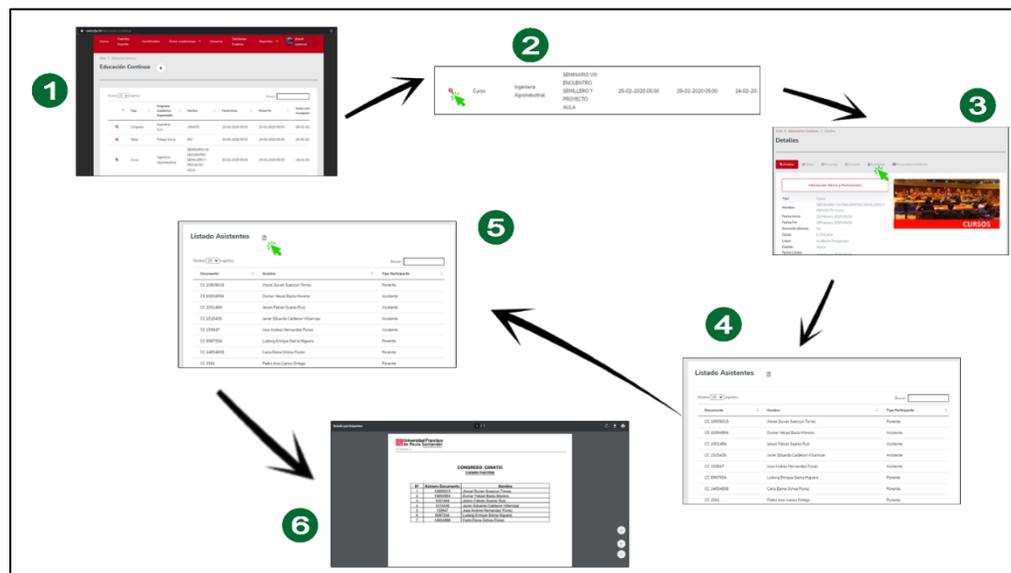


Figura 134 Resultado caso de prueba descargar listado

## ANEXO 4. ITERACIÓN 4 FASE DE DESARROLLO

## ANEXO 4-1. HU9 Generar certificados de asistencia

## Planeación de tareas en tarjeta Kanban

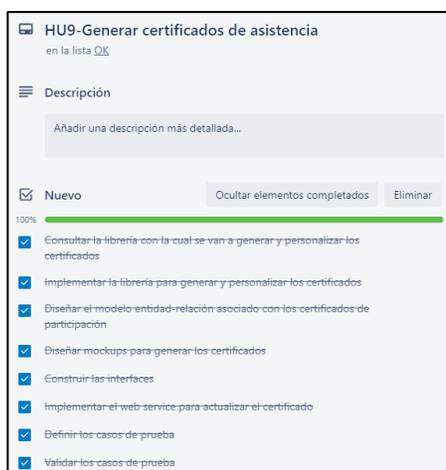


Figura 135 Tarjeta Kanban generar certificados de asistencia.

## Definición del modelo de datos

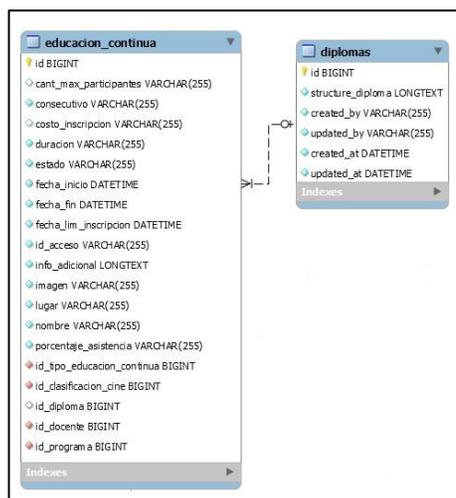


Figura 136 Modelo Entidad-Relación certificados

## Diseño de mockups

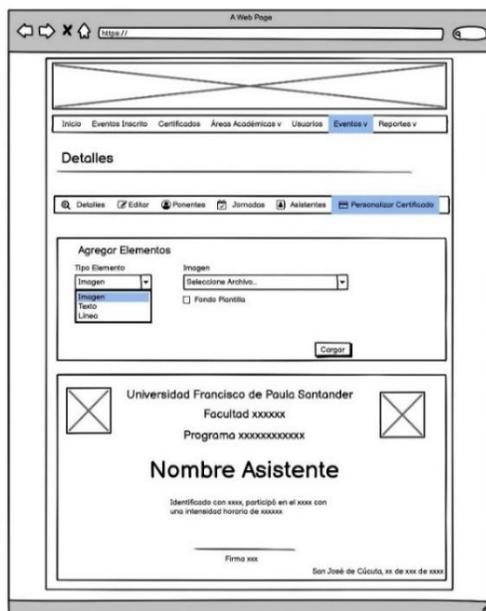


Figura 137 Mockup generar certificado diploma, insertar imágenes.

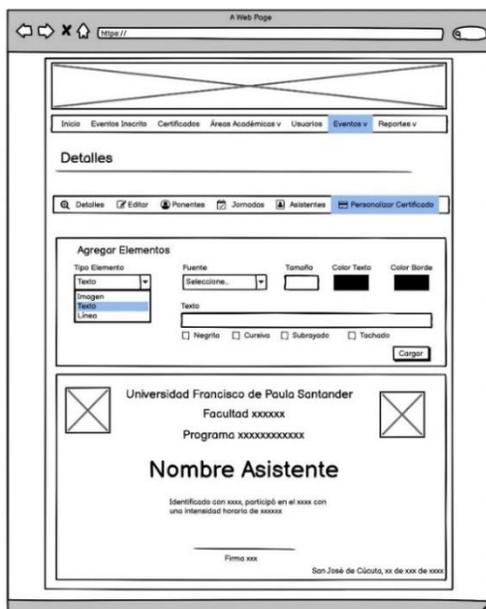


Figura 138 Mockup generar certificado diploma, insertar texto.

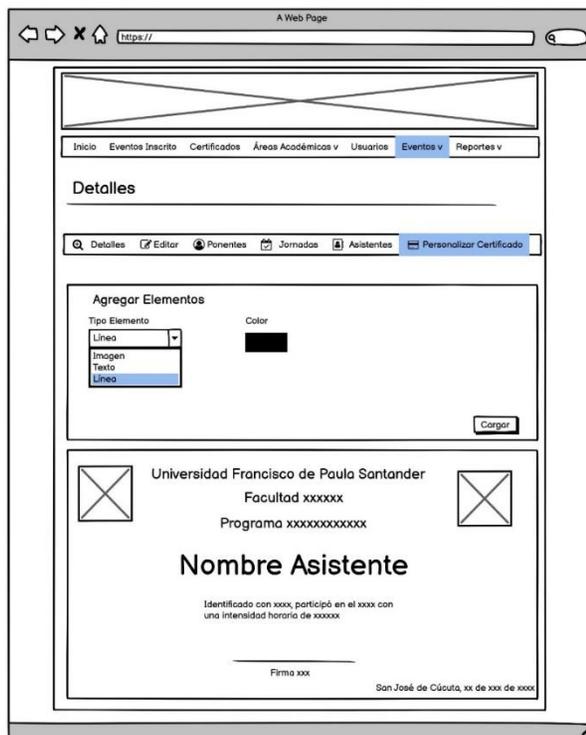


Figura 139 Mockup generar certificado diploma, linea

## Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 81 Caso de prueba diseñar diploma de participación

<b>CASO DE PRUEBA</b>	<b>CP-026</b>
	<b>HU9</b>
	<b>19/04/2021</b>
<b>Nombre</b> Diseñar diploma de participación	
<b>Descripción</b> El usuario crea el diploma de asistencia que los participantes descargarán agregándole el fondo, logos, firmas y titulo/subtítulos según sea el caso.	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> </ul>	
<b>Resultado esperado</b> Actualización exitosa	
<b>Desarrollo</b>	

1. Ubicarse en la interfaz de gestionar eventos
2. Dar click en el botón  para visualizar detalles de la educación continua
3. Dar click en la opción “Personalizar certificado”
4. Personalizar el diploma con elementos, tales como, logos, firmas, titulo
5. Dar click en el botón “Guardar”

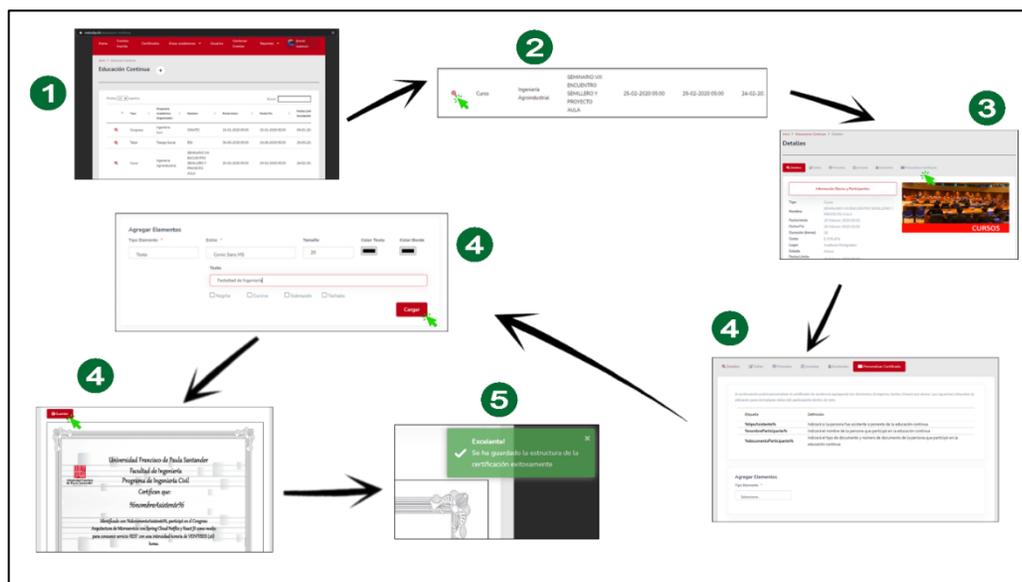


Figura 140 Resultado caso de prueba diseñar diploma de participación.

## ANEXO 4-2. HU10: Obtener certificados de asistencia

### Planeación de tareas en tarjeta Kanban

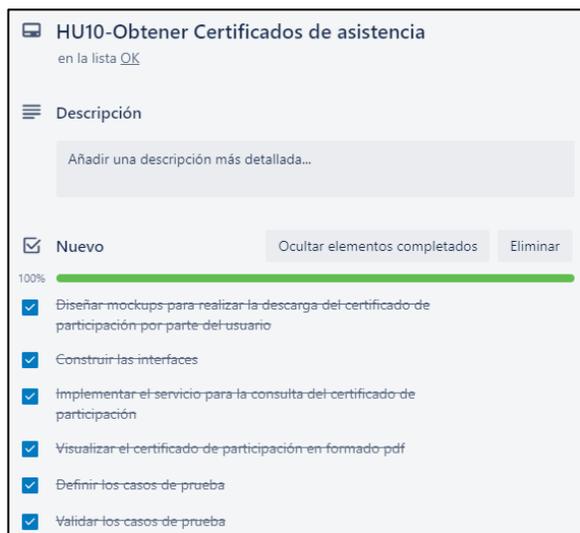


Figura 141 Tarjeta Kanban obtener certificados de asistencia

### Definición del modelo de datos



Figura 142 Modelo Entidad-Relación participantes

## Diseño de mockups

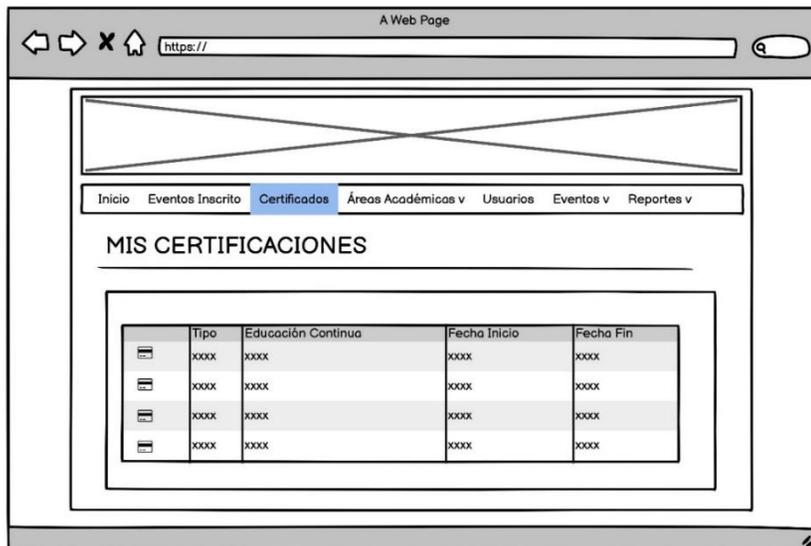


Figura 143 Mockup listado certificaciones.

## Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 82 Caso de prueba descarga de diploma de participación por parte del usuario

CASO DE PRUEBA	CP-027
	HU10
	19/04/2021
<b>Nombre</b> Descarga de diploma de participación por parte del usuario	
<b>Descripción</b> El usuario descarga su diploma de participación en formato .pdf de un curso o evento de educación continua.	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El curso o evento de educación continua debe haber finalizado</li> <li>• El usuario debe haber asistido a todas las jornadas registradas</li> </ul>	
<b>Resultado esperado</b> Descarga exitosa	
<b>Desarrollo</b>	

1. Dar click en el menú “Certificados”
2. Visualizar el listado de todas las educaciones continuas en las cuales ha participado
3. Dar click en el icono  para visualizar el certificado
4. Visualizar el pdf generado con el certificado de participación

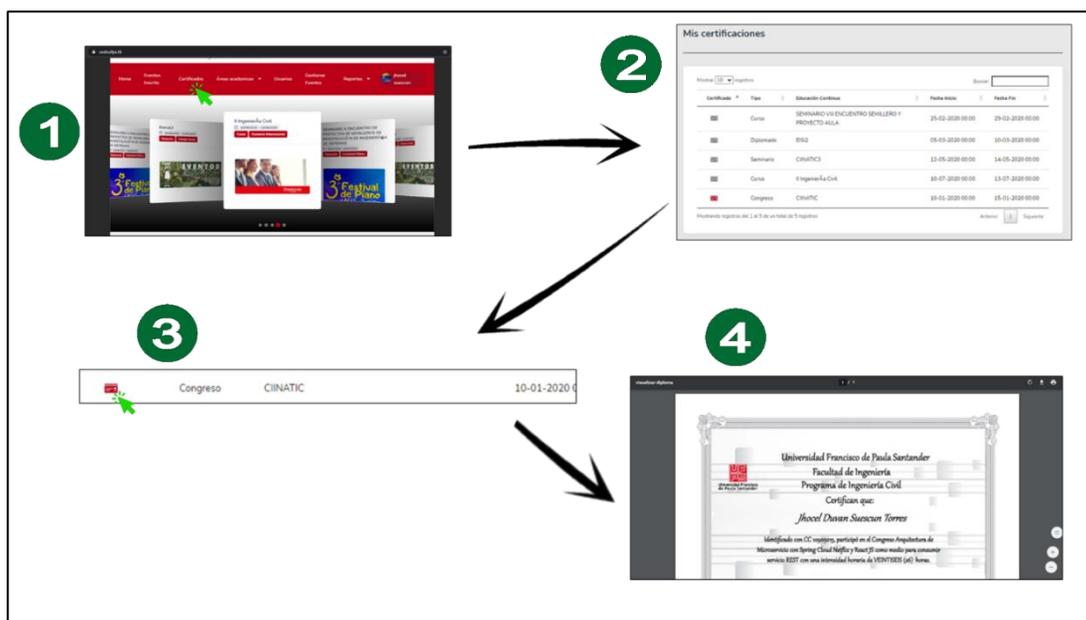


Figura 144 Resultado caso de prueba descarga diploma de participación.

## ANEXO 4-3. HU12 Gestionar asistencia de asistentes inscritos

### Planeación de tareas en tarjeta Kanban

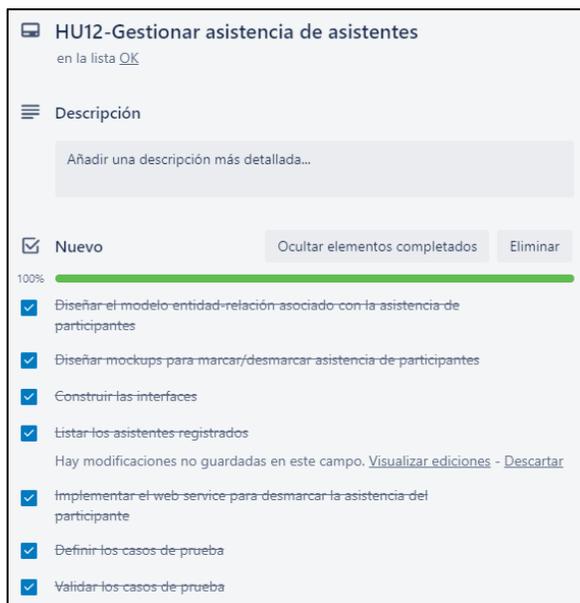


Figura 145 Tarjeta Kanban gestionar asistencia

### Definición del modelo de datos

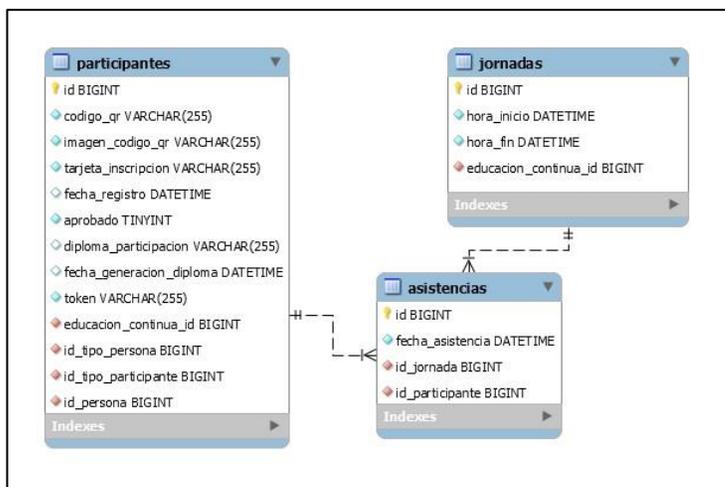


Figura 146 Modelo Entidad-Relación asistencias.

## Diseño de mockups

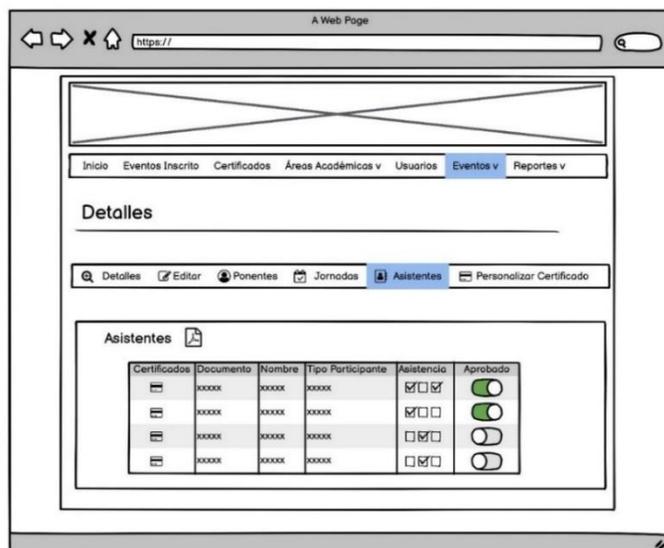


Figura 147 Mockup toma de asistencias

## Definición y validación de casos de pruebas

- Pruebas unitarias

```

@Mock
private IAsistenciaService asistenciaService;

@InjectMocks
private AsistenciaRestController asistenciaRestController;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);
}

@Test
public void marcarAsistenciaGeneralTest() {
    final ResponseEntity?> response = asistenciaRestController.marcarAsistenciaGeneral(1L);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void desmarcarAsistenciaGeneralTest() {
    final ResponseEntity?> response = asistenciaRestController.desmarcarAsistenciaGeneral(1L);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void marcarAsistenciaParticipanteTest() {
    final ResponseEntity?> response = asistenciaRestController.marcarAsistenciaParticipante(1L);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

@Test
public void desmarcarAsistenciaParticipanteTest() {
    final ResponseEntity?> response = asistenciaRestController.desmarcarAsistenciaParticipante(1L);
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}

```

Figura 148 Test toma de asistencias

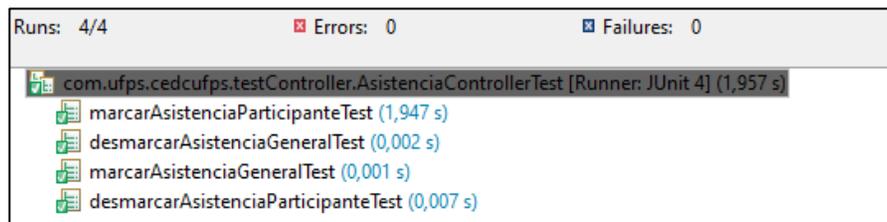


Figura 149 Resultados test toma de asistencias

- Pruebas de aceptación

Tabla 83 Caso de prueba marcar asistencia en una jornada

<b>CASO DE PRUEBA</b>	<b>CP-029</b>
	<b>HU12</b>
	<b>19/04/2021</b>
<b>Nombre</b> Marcar asistencia de un asistente en una jornada	
<b>Descripción</b> El usuario chequea asistencia de un participante en una jornada del curso o evento de educación continua.	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> <li>• La educación continua debe tener mínimo una jornada registrada</li> <li>• La educación continua debe haber iniciado</li> </ul>	
<b>Resultado esperado</b> Actualización exitosa	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de gestionar eventos</li> <li>2. Dar click en el botón  para visualizar detalles de la educación continua</li> <li>3. Dar click en la opción “Asistentes”</li> <li>4. Visualizar listado de asistentes inscritos en la educación continua</li> <li>5. Checkear la asistencia del participante en una jornada y visualizar el mensaje exitoso</li> </ol>	

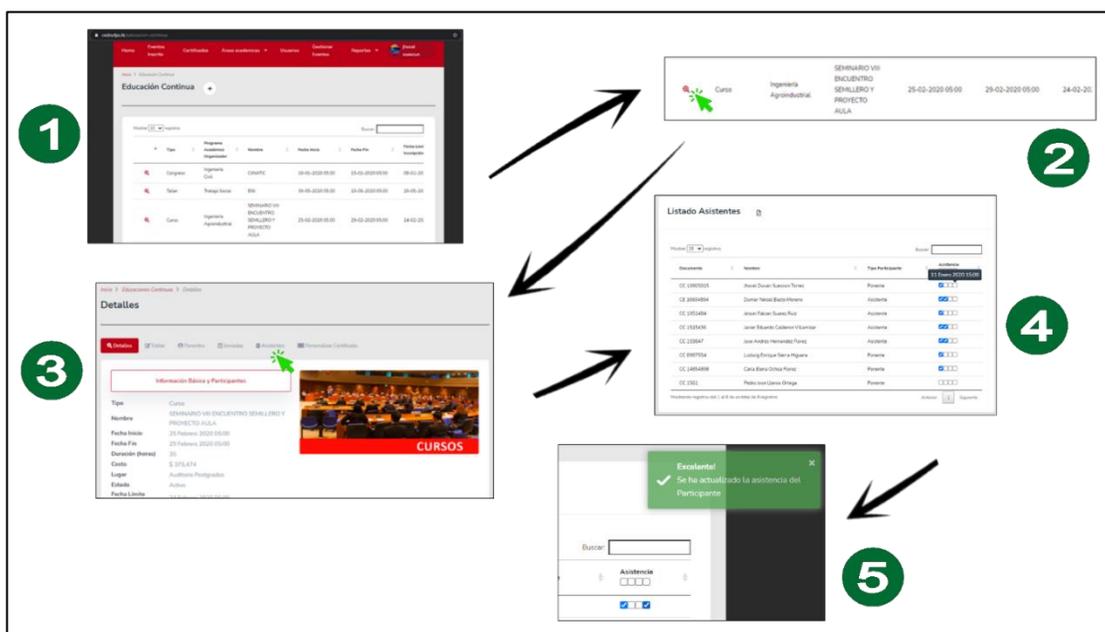


Figura 150 Resultado caso de prueba marcar asistencia de un asistente en una jornada.

Tabla 84 Caso de prueba desmarcar asistencia en una jornada

<b>CASO DE PRUEBA</b>	<b>CP-030</b>
	<b>HU12</b>
	<b>19/04/2021</b>
<b>Nombre</b> Desmarcar asistencia de un asistente en una jornada	
<b>Descripción</b> El usuario retira asistencia de un participante en una jornada del curso o evento de educación continua.	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol: ROLE_MANAECCU</li> <li>• El curso o evento de educación continua debe tener mínimo una jornada registrada</li> <li>• La educación continua debe haber iniciado</li> </ul>	
<b>Resultado esperado</b> Actualización exitosa	
<b>Desarrollo</b> 1. Ubicarse en la interfaz de gestionar eventos	

2. Dar click en el botón  para visualizar detalles de la educación continua
3. Dar click en la opción “Asistentes”
4. Visualizar listado de asistentes inscritos en la educación continua
5. Quitar el check de la asistencia del participante en una jornada y visualizar el mensaje exitoso

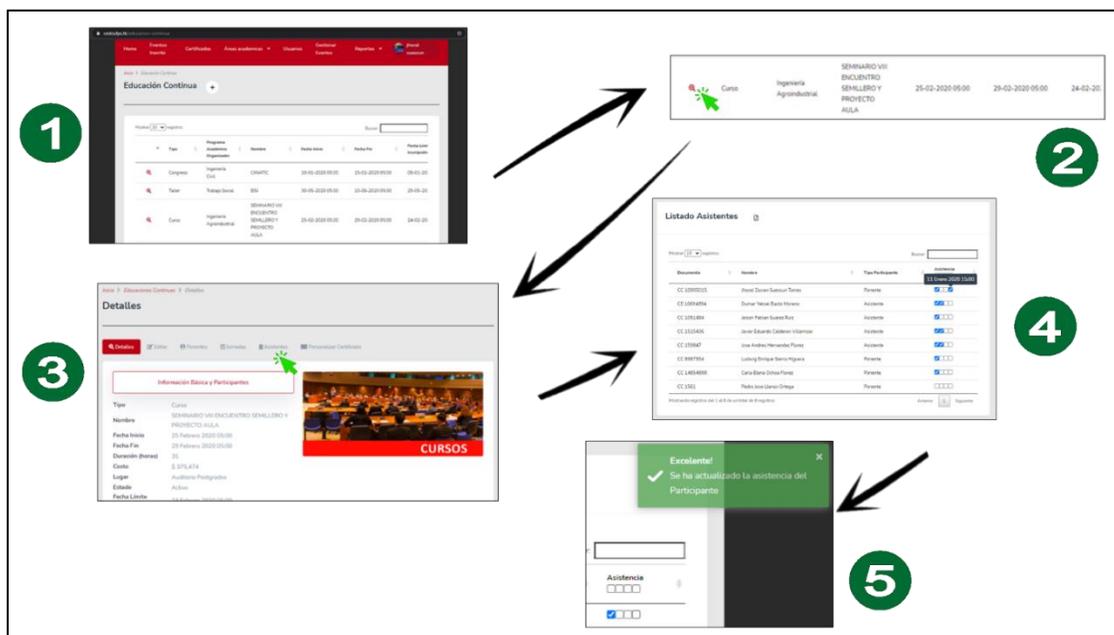


Figura 151 Resultado caso de prueba desmarcar asistencia de un asistente en una jornada



Definición del modelo de datos

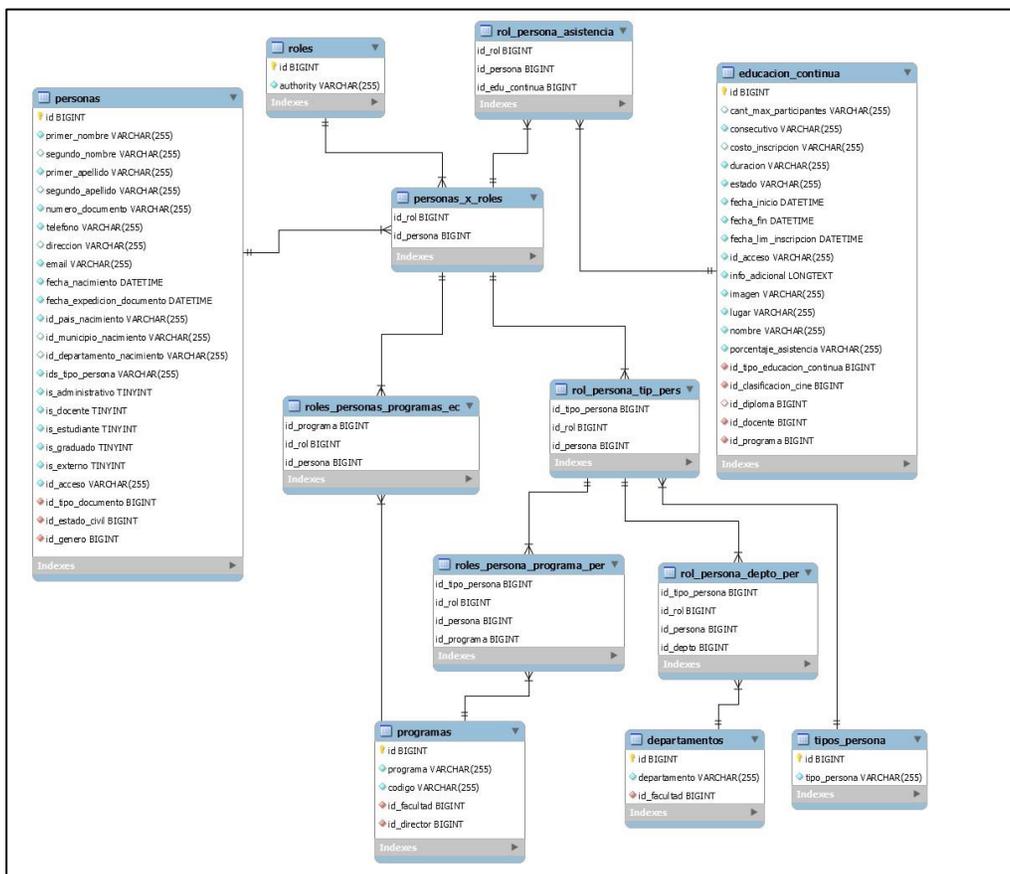


Figura 153 Modelo Entidad-Relación roles usuarios

Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 85 Caso de prueba asignar permisos a un usuario

<b>CASO DE PRUEBA</b>	<b>CP-018</b>
	<b>HU5</b>
	<b>19/04/2021</b>
<b>Nombre</b>	
Asignar permisos a un usuario	
<b>Descripción</b>	

El usuario otorga/revoca roles a algún usuario
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin o ser director de programa</li> </ul>
<b>Resultado esperado</b> Actualización exitosa
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Ubicarse en la interfaz de usuarios</li> <li>2. Dar click en el botón  del registro del usuario para desplegar el formulario</li> <li>3. Asignar los respectivos permisos del usuario</li> <li>4. Dar click en guardar</li> <li>5. Visualizar mensaje de actualización de roles exitosa</li> </ol>

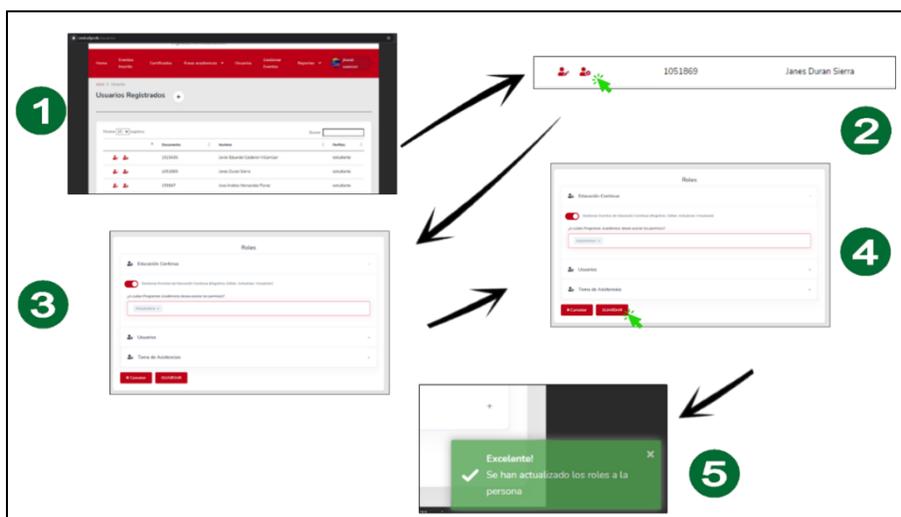


Figura 154 Resultados caso de prueba asignar permisos a un usuario

## ANEXO 5-2. HU8 Visualizar datos estadísticos

### Planeación de tareas en tarjeta Kanban

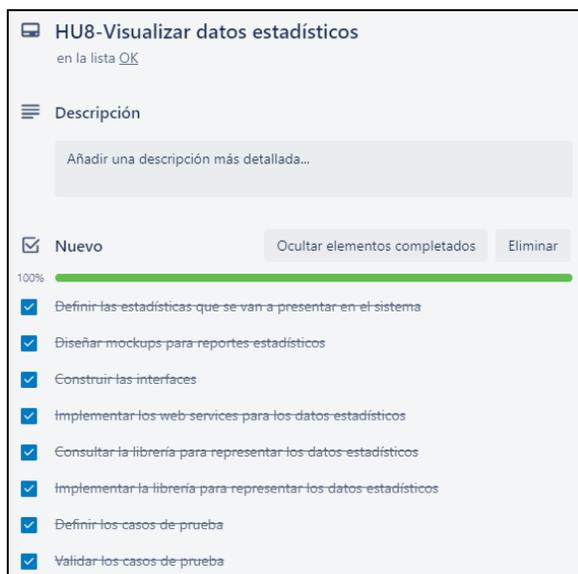


Figura 155 Tarjeta Kanban visualizar datos estadísticos.

### Diseño de mockups

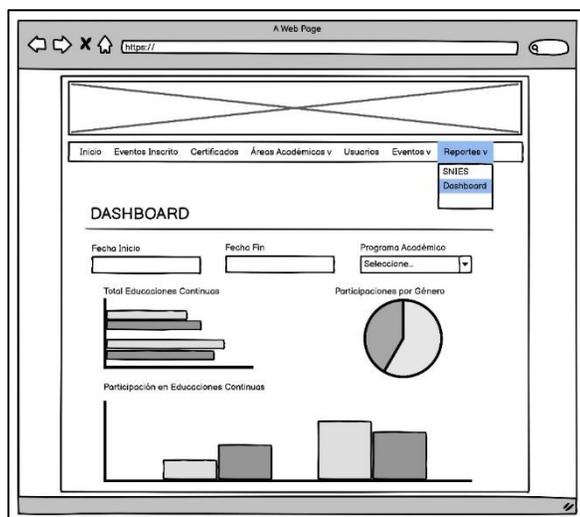


Figura 156 Mockup datos estadísticos.

## Definición y validación de casos de pruebas

- Pruebas Integración

```

@Mock
private IInformeSniesService informeSniesService;

@InjectMocks
private ReportesSniesRestController reporteSniesController;

private static InformeSniesDto INFORME_SNIES_DTO;
private static String FECHA_INICIO;
private static String FECHA_FIN;
private static String DESCRIPCION;
private static List<Object[]> RESULT_STATISTICS_CONTEO_GRAL;
private static List<Object[]> RESULT_STATISTICS_CONTEO_PERSONAS;
private static List<Object[]> RESULT_STATISTICS_CONTEO_GENERO;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);

    RESULT_STATISTICS_CONTEO_GRAL=new ArrayList<Object[]>();
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Educación Continua", "Total"});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Curso",0});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Taller",0});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Seminario",0});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Congreso",0});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Simposio",0});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Diplomado",0});
    RESULT_STATISTICS_CONTEO_GRAL.add(new Object[] {"Otro",0});

    RESULT_STATISTICS_CONTEO_PERSONAS=new ArrayList<Object[]>();
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Educación Continua",
        "Estudiante", "Docente", "Administrativo", "Graduado", "Externo"});
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Curso",0,0,0,0,0});
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Taller",0,0,0,0,0});
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Seminario",0,0,0,0,0});
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Congreso",0,0,0,0,0});
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Simposio",0,0,0,0,0});
    RESULT_STATISTICS_CONTEO_PERSONAS.add(new Object[] {"Diplomado",0,0,0,0,0});
}

```

Figura 157 Instanciación de variables test datos estadísticos.

```

@Test
public void estadisticasConteoGeneralEduContinuaTest() {
    Mockito.when(informeSniesService.generarStatisticsConteoGeneralEduContinua
        (FECHA_INICIO, FECHA_FIN,"1")).thenReturn(RESULT_STATISTICS_CONTEO_GRAL);
    final ResponseEntity<List<Object[]>> response = reporteSniesController
        .estadisticasConteoGeneralEduContinua(FECHA_INICIO, FECHA_FIN,"1");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
    assertFalse(response.getBody().isEmpty());
}

@Test
public void estadisticasConteoGeneralEduContinuaPersonasTest() {
    Mockito.when(informeSniesService.generarStatisticsConteoGeneralPersonas
        (FECHA_INICIO, FECHA_FIN,"1")).thenReturn(RESULT_STATISTICS_CONTEO_PERSONAS);
    final ResponseEntity<List<Object[]>> response = reporteSniesController
        .estadisticasConteoGeneralEduContinuaPersonas(FECHA_INICIO, FECHA_FIN,"1");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
    assertFalse(response.getBody().isEmpty());
}

@Test
public void estadisticasConteoGeneralEduContinuaGenerosTest() {
    Mockito.when(informeSniesService.generarStatisticsConteoGeneralGenero
        (FECHA_INICIO, FECHA_FIN,"1")).thenReturn(RESULT_STATISTICS_CONTEO_GENERO);
    final ResponseEntity<List<Object[]>> response = reporteSniesController
        .estadisticasConteoGeneralEduContinuaGeneros(FECHA_INICIO, FECHA_FIN,"1");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
    assertFalse(response.getBody().isEmpty());
}
}

```

Figura 158 Test datos estadísticos educaciones continuas

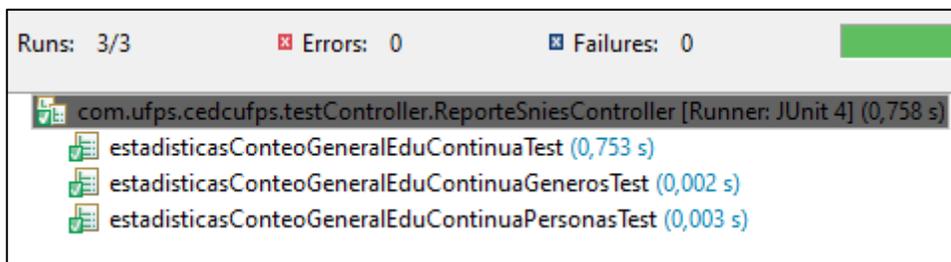


Figura 159 Resultados test datos estadísticos educaciones continuas.

- Pruebas de aceptación

Tabla 86 Caso de prueba visualizar dashboard con estadísticas

<b>CASO DE PRUEBA</b>	<b>CP-025</b>								
	<b>HU8</b>								
	<b>19/04/2021</b>								
<b>Nombre</b>	Visualización dashboard con estadísticas de datos almacenados								
<b>Descripción</b>	El usuario visualiza las estadísticas comparativas de cursos o eventos de educación continua, participantes, asistencias, entre otros.								
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol super-admin o director de programa</li> </ul>								
<b>Resultado esperado</b>	Visualización dashboard								
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>1. Dar click en el menú Reportes, submenú Dashboard</li> <li>2. Ingresar las opciones del filtro para generar las estadísticas <table border="1" style="margin-left: 40px;"> <thead> <tr> <th style="text-align: center;">Campo</th> <th style="text-align: center;">Valor</th> </tr> </thead> <tbody> <tr> <td>Hora Inicio</td> <td>10/6/2020 08:00</td> </tr> <tr> <td>Hora Fin</td> <td>11/12/2020 12:00</td> </tr> <tr> <td>Programa Académico</td> <td>Todos</td> </tr> </tbody> </table> </li> <li>3. Visualizar las estadísticas generadas</li> </ol>	Campo	Valor	Hora Inicio	10/6/2020 08:00	Hora Fin	11/12/2020 12:00	Programa Académico	Todos
Campo	Valor								
Hora Inicio	10/6/2020 08:00								
Hora Fin	11/12/2020 12:00								
Programa Académico	Todos								

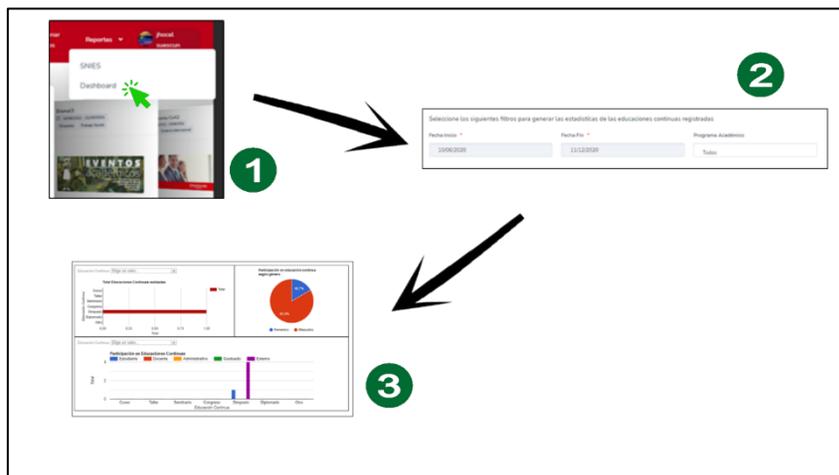


Figura 160 Resultados caso de prueba visualización dashboard

## ANEXO 5-3. HU15 Notificación vía email

## Planeación de tareas en tarjeta Kanban

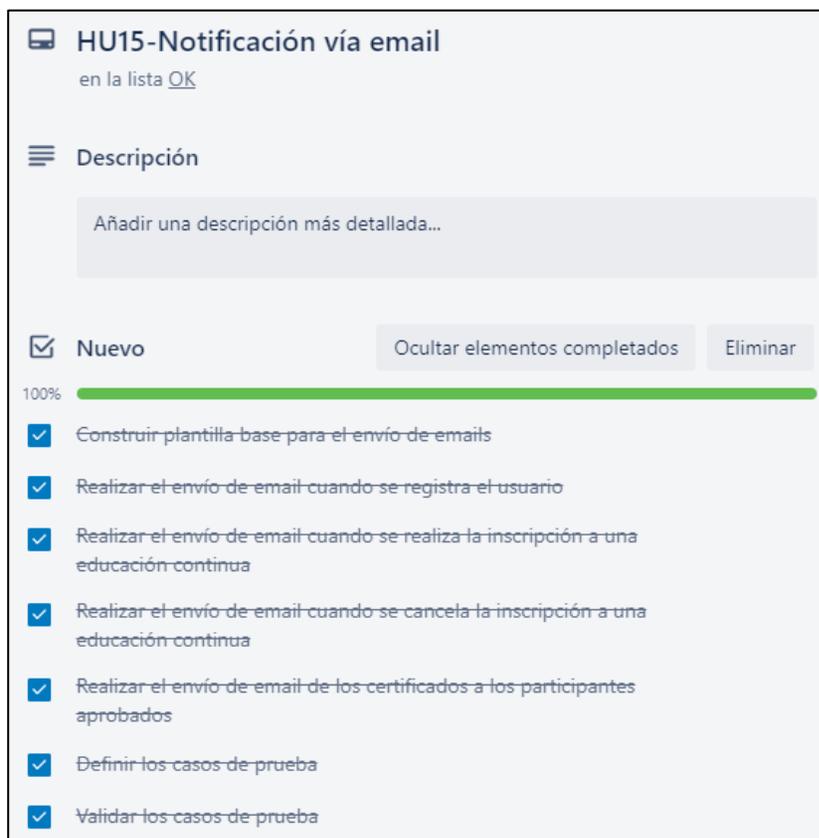


Figura 161 Tarjeta Kanban notificación vía email.

## Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 87 Caso de prueba envío de email de inscripción de curso o evento de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-036</b>
	<b>HU15</b>
	<b>19/04/2021</b>

<b>Nombre</b> Envío de email de inscripción de curso o evento de educación continua
<b>Descripción</b> El usuario recibe un email, en el correo que registró inicialmente, una notificación de inscripción a un curso o evento de educación continua.
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>El usuario debe realizar la inscripción a un curso o evento de educación continua</li> </ul>
<b>Resultado esperado</b> Email recibido
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>El usuario se inscribe a una educación continua</li> <li>El usuario recibe en su bandeja de correo un email que confirma la inscripción exitosa a la educación continua y su respectiva tarjeta de inscripción</li> </ol>

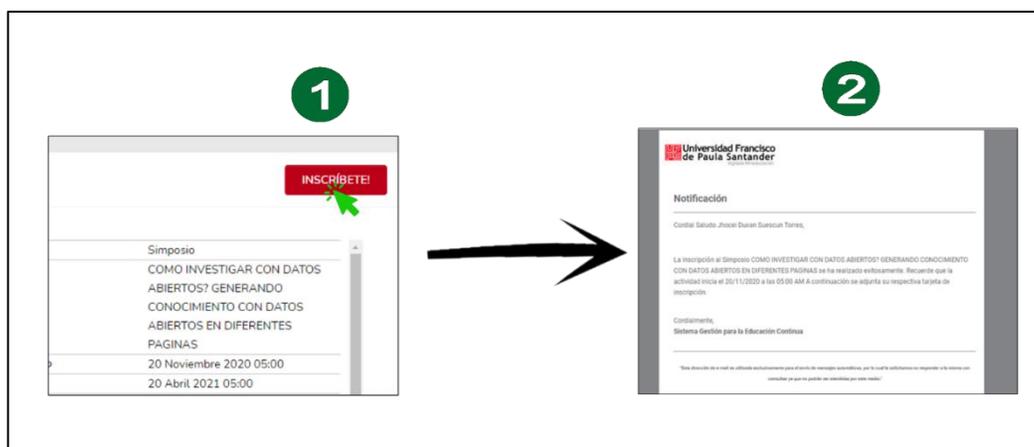


Figura 162 Resultados caso de prueba envío de email de inscripción de curso o evento de educación continua.

Tabla 88 Caso de prueba envío de email de cancelación de inscripción de curso o evento de educación continua

<b>CASO DE PRUEBA</b>	<b>CP-037</b>
	<b>HU15</b>
	<b>19/04/2021</b>
<b>Nombre</b> Envío de email de cancelación de inscripción de curso o evento de educación continua	

<b>Descripción</b>
El usuario recibe un email, en el correo que registró inicialmente, una notificación de cancelación de inscripción a un curso o evento de educación continua.
<b>Prerrequisitos</b>
<ul style="list-style-type: none"> <li>El usuario debe cancelar la inscripción a un curso o evento de educación continua</li> </ul>
<b>Resultado esperado</b>
Email recibido
<ol style="list-style-type: none"> <li>El usuario cancela su inscripción a una educación continua</li> <li>El usuario recibe en su bandeja de correo un email que confirma la cancelación exitosa de la inscripción a la educación continua</li> </ol>

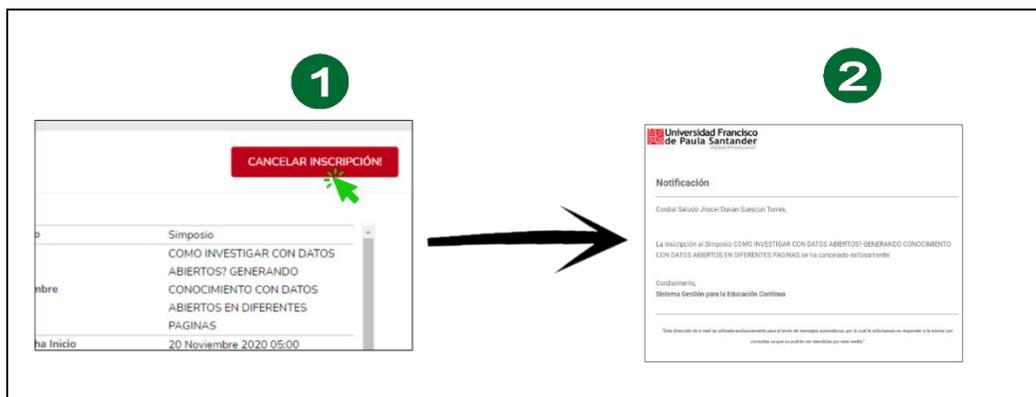


Figura 163 Resultado caso de prueba envío de email de cancelación de inscripción de curso o evento de educación continua.

Tabla 89 Caso de prueba envío de email de registro de nuevo usuario en el sistema

<b>CASO DE PRUEBA</b>	<b>CP-038</b>
	<b>HU15</b>
	<b>19/04/2021</b>
<b>Nombre</b>	
Envío de email de registro de nuevo usuario en el sistema	
<b>Descripción</b>	
El usuario recibe un email, en el correo que registró, una notificación de registro exitoso al sistema.	
<b>Prerrequisitos</b>	
<ul style="list-style-type: none"> <li>No aplica</li> </ul>	
<b>Resultado esperado</b>	

## Email recibido

1. El usuario se registra en el sistema
2. El usuario recibe en su bandeja de correo un email que confirma el registro exitoso en el sistema

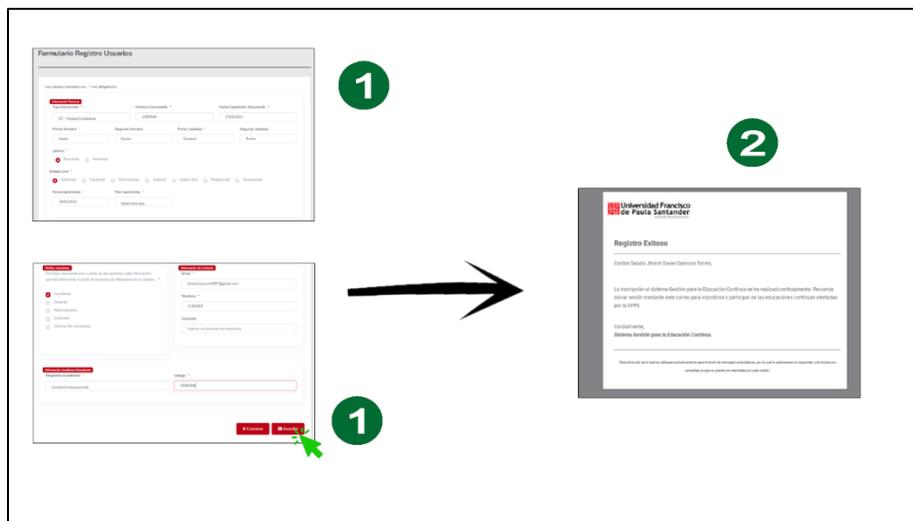


Figura 164 Resultado caso de prueba envío de email de registro de nuevo usuario en el sistema.



## ANEXO 7: Arquitectura de desarrollo hexagonal

### 1) Capa de Infraestructura

Esta capa hace referencia a las configuraciones que la aplicación necesita para funcionar. Se realizó configuración con la base de datos para persistir todos los datos manejados, configuración de email para el envío de emails al momento de realizar el registro al sistema y la inscripción a las educaciones continuas, también se realizó la configuración de autenticación con Google y las configuraciones de seguridad dado los roles del usuario.

- Configuración base de datos

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

Figura 166 Dependencias necesarias para conexión con la base de datos

```
6 spring.datasource.url=jdbc:mysql://3.213.174.86:3306/geduco?serverTimezone=America/Bogota
7 spring.datasource.username=[REDACTED]
8 spring.datasource.password=[REDACTED]
9 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
10 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
```

Figura 167 Configuraciones conexión base de datos

- Configuración envío de correos

```

40 spring.mail.host=smtp.gmail.com
41 spring.mail.port=587
42 spring.mail.protocol=smtp
43 spring.mail.username=[REDACTED]
44 spring.mail.password=[REDACTED]
45 spring.mail.properties.mail.smtp.auth=true
46 spring.mail.properties.mail.smtp.starttls.enable=true
47 spring.mail.properties.mail.smtp.starttls.required=true
48 spring.mail.properties.mail.smtp.quitwait=false

```

Figura 169 Configuración envío de emails

```

@Component
@Configuration
public class EmailConfiguration {

    private EmailProperties emailProperties;

    public EmailConfiguration(EmailProperties emailProperties) {
        this.emailProperties=emailProperties;
    }

    @Bean
    public JavaMailSender getJavaMailSender() {
        JavaMailSenderImpl mailSender = new JavaMailSenderImpl();
        mailSender.setHost(emailProperties.getHost());
        mailSender.setPort(Integer.parseInt(emailProperties.getPort()));
        mailSender.setUsername(emailProperties.getUsername());
        mailSender.setPassword(emailProperties.getPassword());

        Properties props = mailSender.getJavaMailProperties();
        props.put("mail.transport.protocol", "smtp");
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.debug", "true");

        return mailSender;
    }
}

```

Figura 168 Bean de configuración para envío de emails

- Configuración seguridad autenticación

```
spring.security.oauth2.client.registration.google.client-id=[REDACTED]
spring.security.oauth2.client.registration.google.client-secret=[REDACTED]

spring.security.oauth2.appdebug=[REDACTED]
spring.security.oauth2.appproduccion=[REDACTED]
```

Figura 170 Configuración autenticación Google web y App

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    // TODO Auto-generated method stub
    http.authorizeRequests().antMatchers("/", "/politicaprivacidad", "/change-email", "/update-email/**", "/bienvenida", "/re
    .antMatchers("/preinscripcion/**", "/participaciones-educacion-continua/**", "/certificaciones-educacion-continua/**",
    .antMatchers("/facultades/**", "/departamentos-academicos/**", "/programas-academicos/**").hasRole("SUPERADMIN")
    .antMatchers("/educacion-continua/**").hasAnyRole("SUPERADMIN", "MANAECCU")
    .antMatchers("/reportes-SNIES/**").hasAnyRole("SUPERADMIN", "SNIES")
    .antMatchers("/usuarios/**").hasAnyRole("SUPERADMIN", "MANPEOPLE")
    .anyRequest().authenticated()
    .and()
    .logout().logoutSuccessUrl("/logout").logoutSuccessHandler(this.oidcLogoutSuccessHandler()).clearAuthentication(true)
    .and().cors().configurationSource(corsConfigurationSource())
    .and()
    .sessionManagement()
    .sessionCreationPolicy(SessionCreationPolicy.ALWAYS)
    .and()
    .oauth2Login(oauthLogin -> oauthLogin

        .authorizedClientService(this.authorizedClientService())

        .failureHandler(customAuthenticationFailureHandler())
        .successHandler(customSuccessHandler())
        .authorizationEndpoint()
        .authorizationRequestRepository(authorizationRequestRepository()).and()
        .userInfoEndpoint()
        .oidcUserService(this.oidcUserService());
    }
}
```

Figura 171 Configuración de seguridad según roles

## 2) Capa de aplicación

Esta capa hace referencia a la aplicación como tal y los otros programas que interactúan con el nuestro. Debe contener elementos como interfaces de usuario, controladores REST y bibliotecas de serialización.

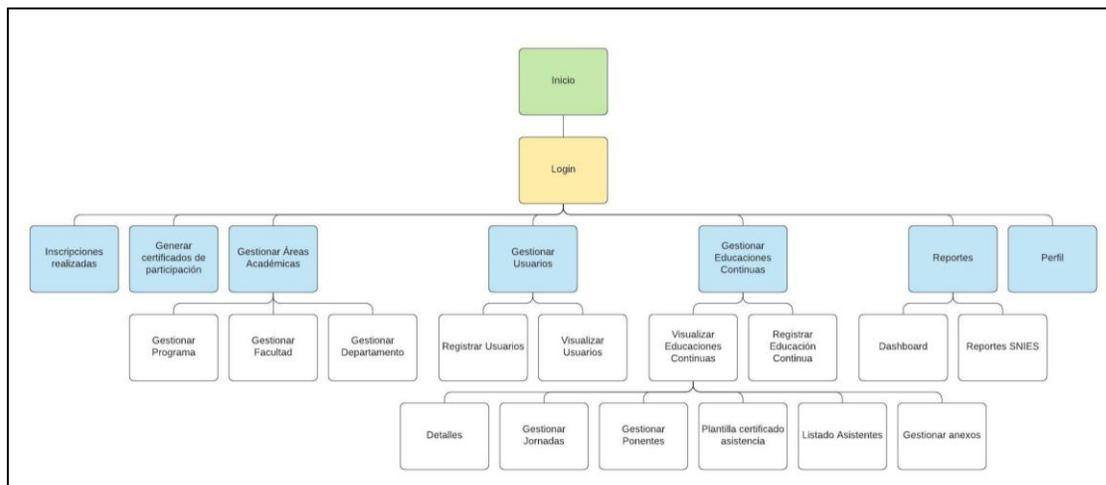


Figura 172 Mapa de navegación de la aplicación web

### 3) Capa de dominio

Es el núcleo de nuestra aplicación, contiene toda la lógica del negocio, es decir, las entidades (Ver Anexo 6), repositorios y servicios utilizados. En las siguientes imágenes, se presentarán diagramas UML de estos ítems.

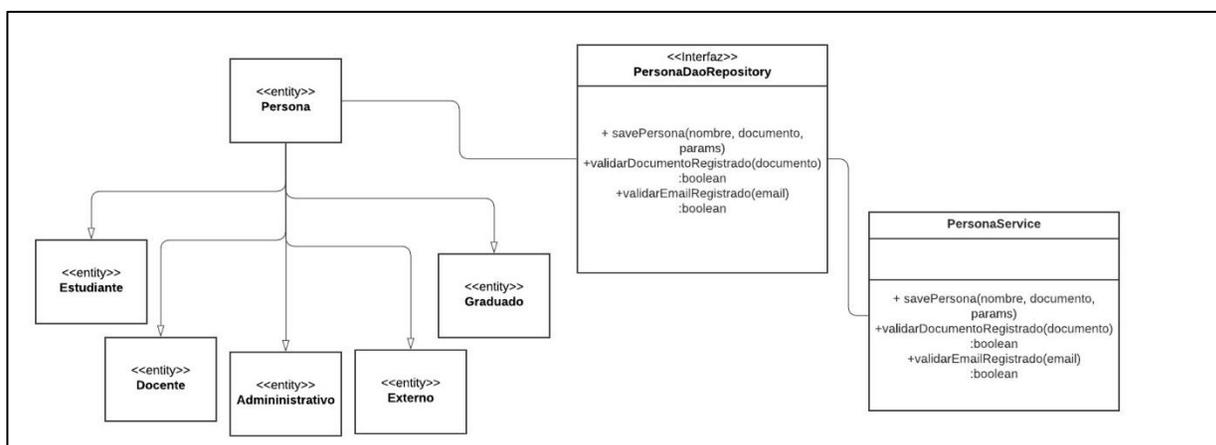


Figura 173 Diagrama UML Persona

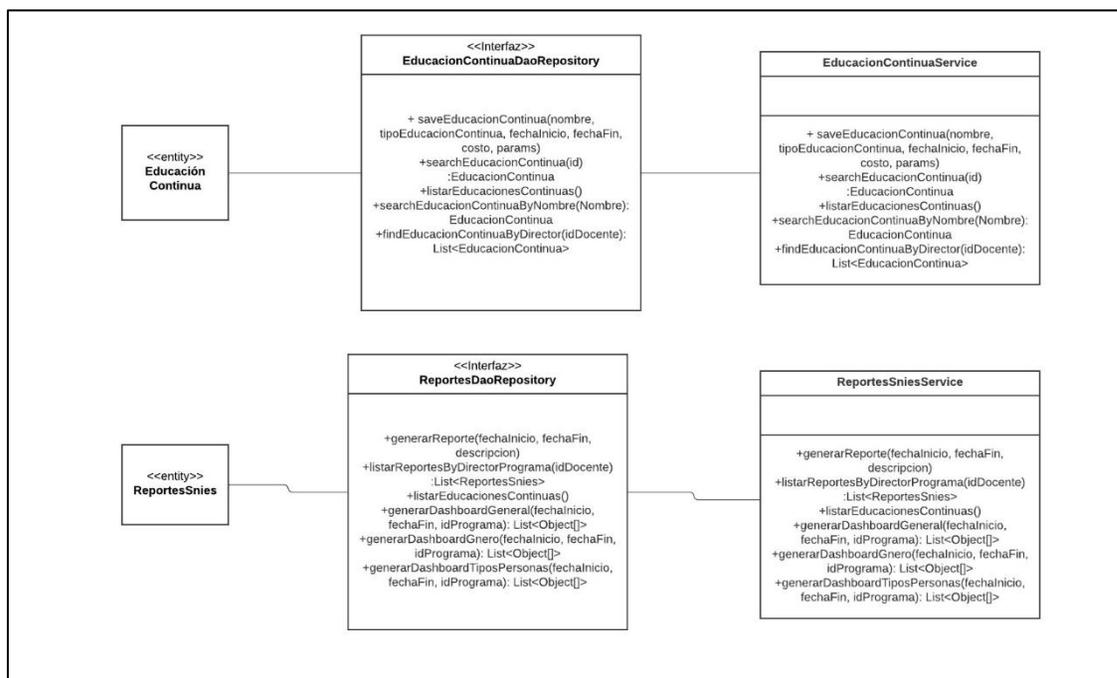


Figura 175 Diagrama UML Educación Continua y Reportes SNIES

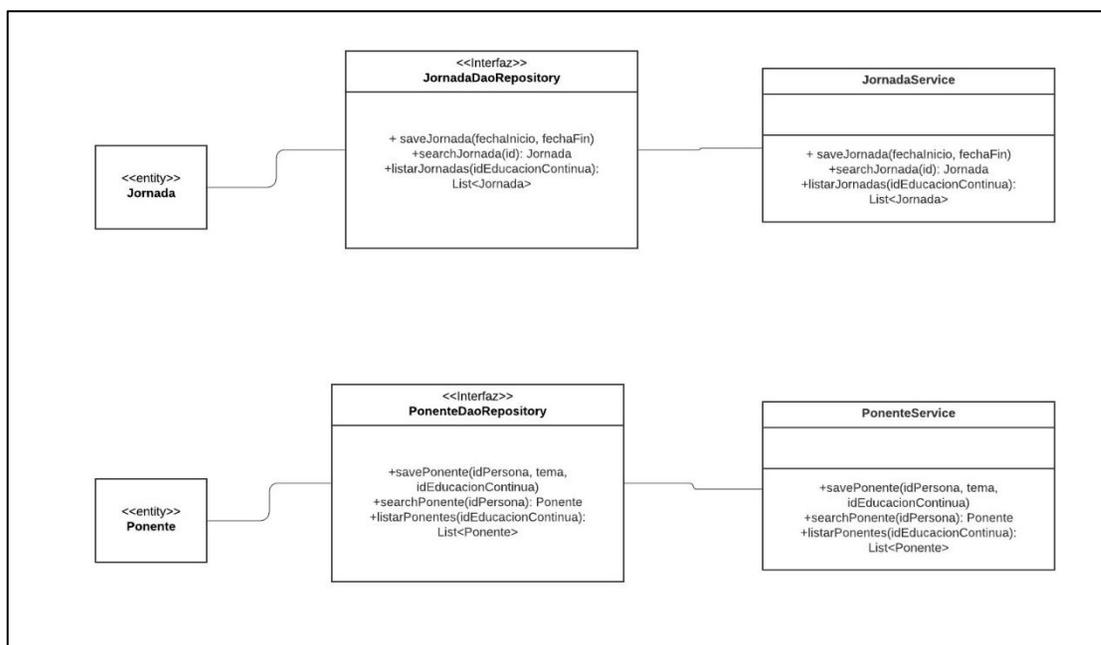


Figura 174 Diagrama UML jornada y ponente

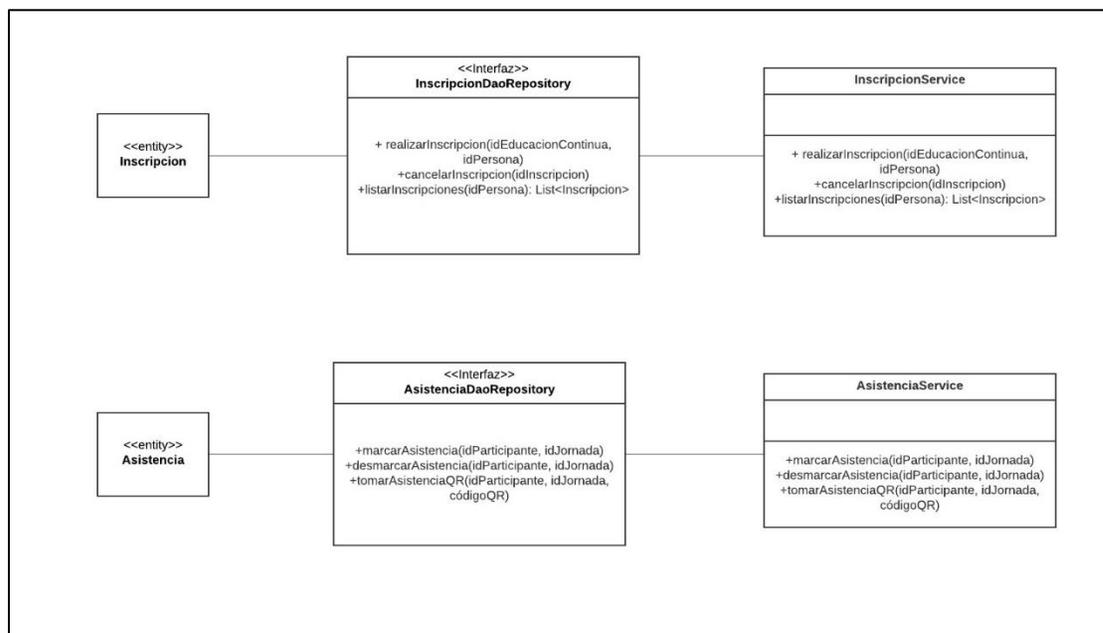


Figura 176 Diagrama UML inscripción y asistencia

## ANEXO 8. Dockerfile sistema gestión educación continua

```
FROM openjdk:8-alpine
```

```
ADD cedcufps.jar cedcufps.jar
```

```
VOLUME /files
```

```
EXPOSE 8080
```

```
ENTRYPOINT ["java", "-jar", "cedcufps.jar", "/bin/bash"]
```

## ANEXO 9. Instalación de *containers* durante la fase de desarrollo

- 1) Crear red en la que se van a conectar los dockers

```
sudo docker network create --driver bridge red
```

- 2) Ejecutar el contenedor Nginx Proxy

```
sudo docker run -d --name nginx-proxy -p 80:80 -p 443:443 -v /etc/nginx/certs -v  
/etc/nginx/vhost.d -v /usr/share/nginx/html -v /var/run/docker.sock:/tmp/docker.sock:ro  
--network red jwilder/nginx-proxy
```

- 3) Ejecutar el contenedor Let's Encrypt

```
sudo docker run --detach \  
--name nginx-proxy-letsencrypt \  
--volumes-from nginx-proxy \  
--volume /var/run/docker.sock:/var/run/docker.sock:ro \  
--env "DEFAULT_EMAIL=jhocelsuescun1997@gmail.com" \  
jrcs/letsencrypt-nginx-proxy-companion
```

Donde se tuvieron en cuenta las siguientes variables de entorno:

DEFAULT\_EMAIL: Es el email al cuál se le va a asignar el certificado de seguridad

## 4) Ejecutar el contenedor MariaDB

```

sudo docker run -d --name mariadb -p 3306:3306 -e
MYSQL_ROOT_PASSWORD=cedcufps -e MYSQL_DATABASE=tesis -e
MYSQL_USER=user_tesis -e MYSQL_PASSWORD=pass_user -v /home/ec2-
user/tesis/db:/var/lib/mysql --network red mariadb

```

Donde se tuvieron en cuenta las siguientes variables de entorno:

MYSQL\_ROOT\_PASSWORD: Es la contraseña que tendrá el admin de la BD root

DATABASE: Es el nombre de la base de datos utilizada para guardar los datos del sistema gestión para la educación continua

MYSQL\_USER: Es el nombre de usuario para acceder a la base de datos

MYSQL\_PASSWORD: Es la contraseña del usuario creado

El volumen -v /home/ec2-user/tesis/db:/var/lib/mysql indica el directorio donde se van a guardar los archivos de la base de datos

## 5) Ejecutar el contenedor PhpMyAdmin

```

sudo docker run -d --name phpmyadmin -e PMA_HOST=mariadb -e
PMA_PORT=3306 -e
VIRTUAL_HOST=ufpsphpmyadmin.ml,www.ufpsphpmyadmin.ml -e
LETSENCRYPT_HOST=ufpsphpmyadmin.ml,www.ufpsphpmyadmin.ml -e
LETSENCRYPT_EMAIL=jhocelsuescun1997@gmail.com --network red
phpmyadmin/phpmyadmin

```

Donde se tuvieron en cuenta las siguientes variables de entorno:

PMA\_HOST: Es el nombre del container de motor de base de datos

PMA\_PORT: Es el puerto por el cuál se esta ejecutando el motor de base de datos

VIRTUAL\_HOST y LETSENCRYPT\_HOST: Es el dominio creado para acceder al phpmyadmin

LETSENCRYPT\_EMAIL: Es el correo ingresado en el segundo contenedor para el certificado de seguridad

6) Ejecutar el contenedor cedcufps

```
sudo docker build -t cedcufps .
```

```
sudo docker run -d -v /home/ec2-user/cedcufps/files:/files -e
```

```
SPRING_DATASOURCE_URL=jdbc:mysql://34.239.226.222:3306/tesis?serverTimezo
```

```
ne=America/Bogota -e SPRING_DATASOURCE_USERNAME=root -e
```

```
SPRING_DATASOURCE_PASSWORD=cedcufps -e
```

```
VIRTUAL_HOST=cedcufps.tk,www.cedcufps.tk -e
```

```
LETSENCRYPT_HOST=cedcufps.tk,www.cedcufps.tk -e
```

```
LETSENCRYPT_EMAIL=jhocelsuescun1997@gmail.com --network red cedcufps
```

Donde se tuvieron en cuenta las siguientes variables de entorno:

**SPRING\_DATASOURCE\_URL:** Url de acceso a la base de datos

**SPRING\_DATASOURCE\_USERNAME:** Es el usuario que va a acceder a la base de datos

**SPRING\_DATASOURCE\_PASSWORD:** Es la contraseña del usuario que va a acceder a la base de datos

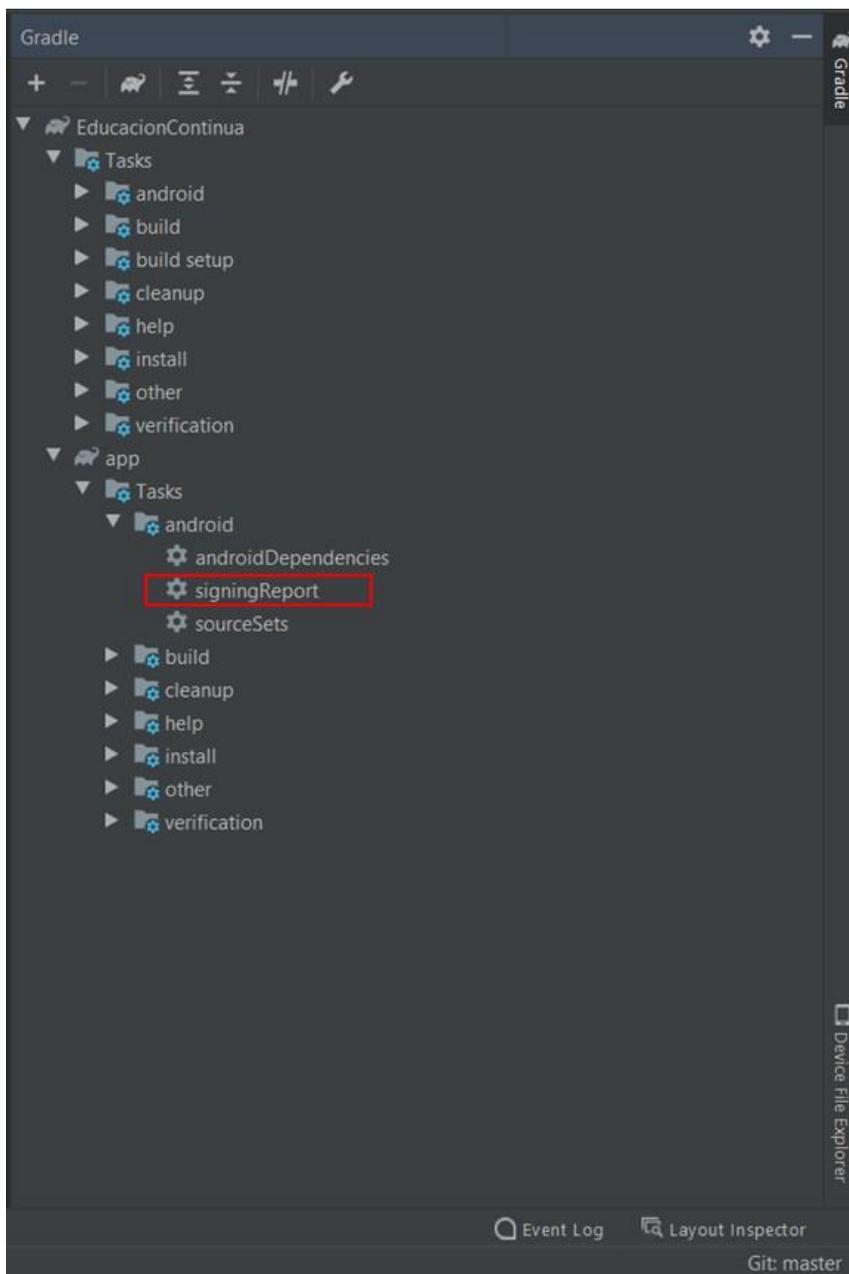
**VIRTUAL\_HOST y LETSENCRYPT\_HOST:** Es el dominio creado para acceder al sistema gestión de educación continua

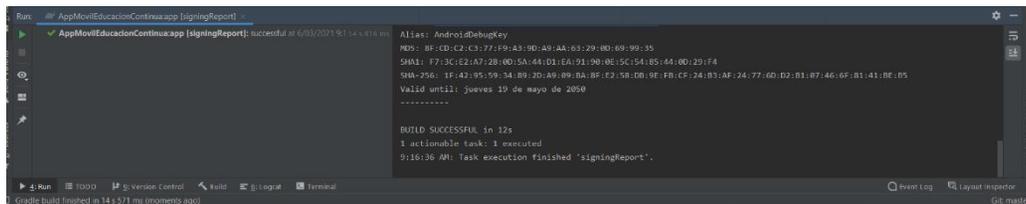
**LETSENCRYPT\_EMAIL:** Es el correo ingresado en el segundo contenedor para el certificado de seguridad

## ANEXO 10. ITERACIÓN 3 FASE DE DESARROLLO

## ANEXO 10-1. Implementación Login Google en app móvil

- 1) Identificar la key sha1 para debug que se encuentra en el archivo signingReport del proyecto en Android Studio como se indica en la siguiente imagen





- 2) Con el siguiente comando se genera la key para producción

*keytool -keystore path-to-debug-or-production-keystore -list -v*

```
Valid from: Sun Aug 23 02:49:08 COT 2020 until: Thu Aug 17 02:49:08 COT 2045
Certificate fingerprints:
  SHA1: AD:22:D0:F7:CB:EB:45:81:F6:30:75:51:B3:59:0C:29:BA:6E:05:E4
  SHA256: F7:57:74:BD:77:A2:1F:F5:3C:94:86:78:81:09:13:83:8C:C2:12:CD:71:9A:B3:F6:9D:DF:7F:C3:B4:8C:7F:DB
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3
```

Las keys de debug y producción se utilizarán para generar las credenciales en el api console developers de Google.

- 3) Loguearse en <http://console.developers.google.com/> para realizar las configuraciones
- 4) Crear un proyecto nuevo para el debug de la aplicación



- 5) Completar el formulario de registro para generar el proyecto para el debug de la app móvil

### Proyecto nuevo

 Tienes 19 projects restantes en tu cuota. Solicita un incremento o borra algunos proyectos. [Más información](#)

[MANAGE QUOTAS](#)

Nombre del proyecto \*  
appcedcufpsdebug 

ID de proyecto: appcedcufpsdebug.No se podrá cambiar más tarde. [EDITAR](#)

Organización \*  
ufps.edu.co 

Selecciona una organización para vincularla a un proyecto. No podrás cambiar esta selección más adelante.

Ubicación \*  
 ufps.edu.co [EXPLORAR](#)

Organización o carpeta superior

[CREAR](#) [CANCELAR](#)

- 6) Dar click en la opción “credenciales” del menú lateral



7) Dar click en crear credenciales, opción “ID de cliente de OAuth”



8) Diligenciar el formulario para generar la credencial del login para debug ingresando el SHA-1 respectivo de debug

← Crear ID de cliente de OAuth

Un ID de cliente se usa con el fin de identificar una sola app para los servidores de OAuth de Google. Si la app se ejecuta en varias plataformas, cada una necesitará su propio ID de cliente. Consulta [Configura OAuth 2.0](#) para obtener más información.

Tipo de aplicación \*  
Android

[Más información](#) sobre los tipos de clientes de OAuth

Nombre \*  
Cliente app debug

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

Nombre del paquete \*  
com.example.educacioncontinua

En tu archivo AndroidManifest.xml

Huella digital del certificado SHA-1 \*  
F7:3C:E2:A7:2B:0D:5A:44:D1:EA:91:90:0E:5C:54:85:44:0D:29:F4

El certificado de firma SHA-1 con huella digital restringe el uso de tus aplicaciones para Android. [Más información](#)

Use this command to get the fingerprint

9) Posterior a esto se autogenera una key web client que utilizaremos para realizar la configuración del login de la aplicación mediante rest service.

ID de clientes OAuth 2.0				
<input type="checkbox"/>	Nombre	Fecha de creación ↓	Tipo	ID de cliente
<input type="checkbox"/>	OAuth client	15 nov. 2020	Android	877594372856-ek91...
<input type="checkbox"/>	Web client (Auto-created for Google Sign-in)	15 nov. 2020	Aplicación web	877594372856-aegs...

10) Realizamos el mismo proceso para generar las keys para producción

11) Una vez creada, tomamos el ID de cliente del Web cliente auto creado en debug y producción y lo agregamos a las propiedades del proyecto de spring boot

```
spring.security.oauth2.appdebug=877594372056-aegs95db4kk73knee04esc80phfn3lbp.apps.googleusercontent.com  
spring.security.oauth2.appproduccion=877594372056-aegs95db4kk73knee04esc80phfn3lbp.apps.googleusercontent.com
```

Una vez configurado el proyecto, se realiza la codificación y las pruebas pertinentes para validar el funcionamiento correcto del login a través de Google en la aplicación móvil y el funcionamiento de los requerimientos.

## ANEXO 10-2. HU-A1 Login de la app

## Planeación de tareas en tarjeta Kanban

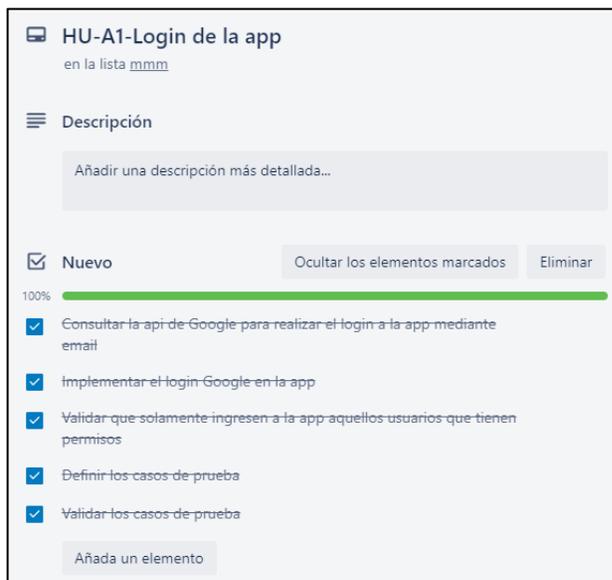


Figura 177 Tarjeta Kanban login app móvil

## Diseño de mockups



Figura 178 Mockup login app

## Definición y validación de casos de pruebas

- Pruebas de aceptación

Tabla 90 Caso de prueba login de app móvil de un usuario con permisos

CASO DE PRUEBA	CP-032
	HU14
	19/04/2021
<b>Nombre</b> Login de app móvil de un usuario con permisos	
<b>Descripción</b> El usuario intenta acceder a la app mediante la cuenta Google	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe tener rol ROLE_SUPERADMIN, ROLE_MANAECCU o ROLE_ATTENDANCE</li> </ul>	
<b>Resultado esperado</b> Inicio de Sesión exitoso	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Abrir app</li> <li>2. Loguearse mediante cuenta Google</li> <li>3. Inicio de Sesión exitoso</li> </ol>	

Nota: Elaboración propia

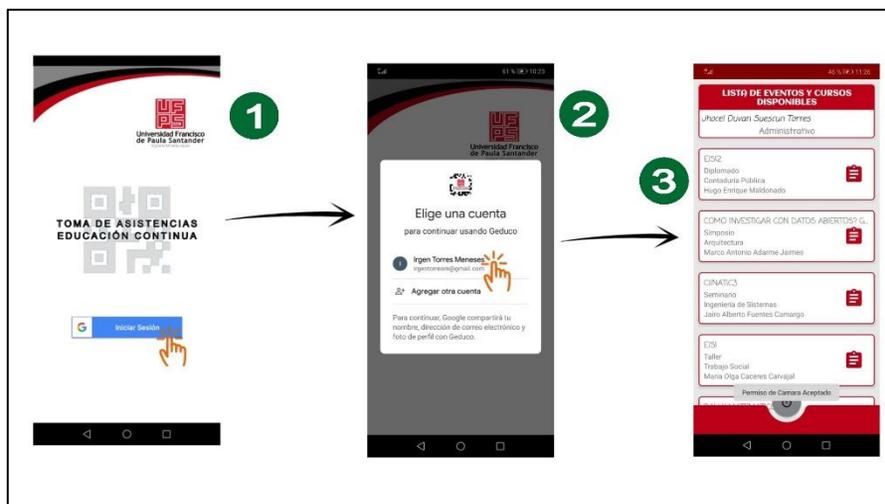


Figura 179 Resultado caso de prueba Login de app móvil de un usuario con permisos.

Tabla 91 Caso de prueba login de app móvil de un usuario sin permisos

CASO DE PRUEBA	CP-033
	HU14
	19/04/2021
<b>Nombre</b>	Login de app móvil de un usuario sin permisos
<b>Descripción</b>	El usuario intenta acceder a la app mediante la cuenta Google pero no es posible debido a que no tiene permisos
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>El usuario no debe tener rol ROLE_SUPERADMIN, ROLE_MANAECCU ni ROLE_ATTENDANCE</li> </ul>
<b>Resultado esperado</b>	Inicio de Sesión fallido
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>1. Abrir app</li> <li>2. Loguearse mediante cuenta Google</li> <li>3. Inicio de Sesión fallido</li> </ol>

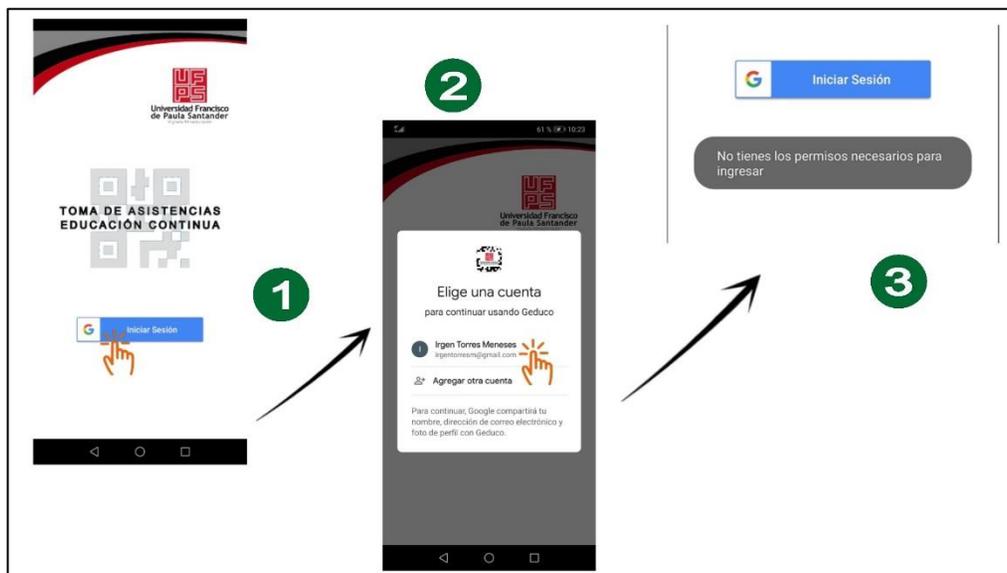


Figura 180 Resultado caso de prueba Login de app móvil de un usuario sin permisos

## ANEXO 10-3. HU-A2 Listar educaciones continuas

### Planeación de tareas en tarjeta Kanban

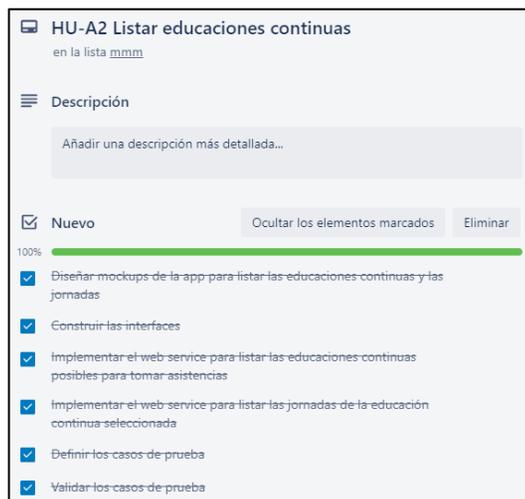


Figura 181 Tarjeta Kanban listar educaciones continuas en aplicación móvil

### Diseño de mockups

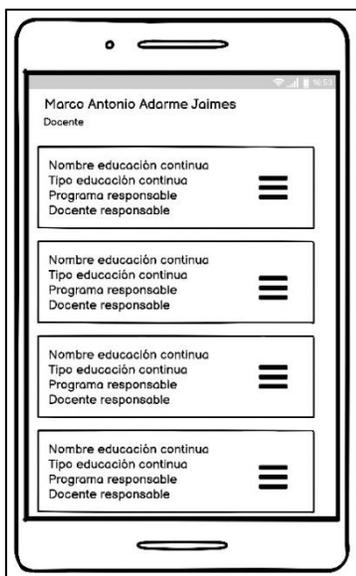


Figura 182 Mockup listar educaciones continuas

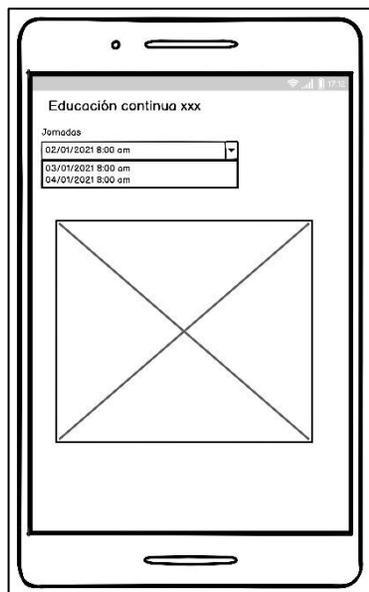


Figura 183 Mockup seleccionar jornadas en app

Definición y validación de casos de pruebas

Pruebas unitarias

```

@Mock
private IEducacionContinuaService educacionContinuaService;

@Mock
private IPersonaService personaService;

@InjectMocks
private AppRestController appRestController;

private static List<EducacionContinuaAppDto> LIST_EDUCACIONES_CONTINUAS;
private static List<JornadaAppDto> JORNADAS;
private static Map<Integer, ParticipanteDto> MAP_PARTICIPANTE;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);

    LIST_EDUCACIONES_CONTINUAS=new ArrayList<EducacionContinuaAppDto>();

    EducacionContinuaAppDto eduContinua1= new EducacionContinuaAppDto();
    eduContinua1.setId(1L);
    eduContinua1.setNombre("CIINATIC");
    eduContinua1.setFechaInicio(new Date());

    EducacionContinuaAppDto eduContinua2= new EducacionContinuaAppDto();
    eduContinua2.setId(2L);
    eduContinua2.setNombre("EISE");
    eduContinua2.setFechaInicio(new Date());

    LIST_EDUCACIONES_CONTINUAS.add(eduContinua1);
    LIST_EDUCACIONES_CONTINUAS.add(eduContinua2);

    JORNADAS= new ArrayList<JornadaAppDto>();

    JornadaAppDto jornada1= new JornadaAppDto();
    jornada1.setId(1L);
    jornada1.setHoraInicio(new Date());
    
```

Figura 184 Instanciación variables test toma asistencias mediante QR.

```

@Test
public void searchCursosYEventos() {
    Mockito.when(educacionContinuaService.findAllEducacionesApp(1L)).
        thenReturn(LIST_EDUCACIONES_CONTINUAS);
    final ResponseEntity<List<EducacionContinuaAppDto>> response = appRestController.
        searchCursosYEventos(1L);
    assertEquals(response.getBody().size(), 2);
}

@Test
public void searchJornadasCursosYEventos() {
    Mockito.when(educacionContinuaService.findAllJornadasByEduContinuaApp(1L)).
        thenReturn(JORNADAS);
    final ResponseEntity<List<JornadaAppDto>> response = appRestController.
        searchJornadasCursosYEventos(1L);
    assertEquals(response.getBody().size(), 2);
}

```

Figura 186 Test buscar educaciones continuas y jornadas

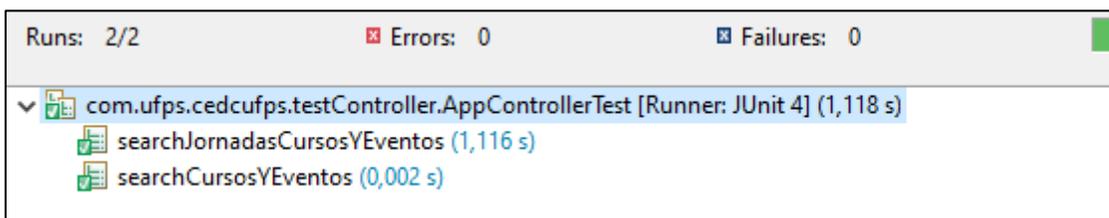


Figura 185 Resultado test buscar educaciones continuas y jornadas

- Pruebas de aceptación

Tabla 92 Caso de prueba login de app móvil de un usuario con permisos

CASO DE PRUEBA	CP-032
	HU14
	19/04/2021
<b>Nombre</b> Listar las educaciones continuas en la app	
<b>Descripción</b> El usuario visualiza las educaciones continuas en desarrollo en la app móvil	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe tener rol ROLE_SUPERADMIN, ROLE_MANAECCU o ROLE_ATTENDANCE</li> </ul>	
<b>Resultado esperado</b>	

Listado de educaciones continuas
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Loguearse mediante cuenta Google</li> <li>2. Visualizar listado de educaciones continuas</li> </ol>

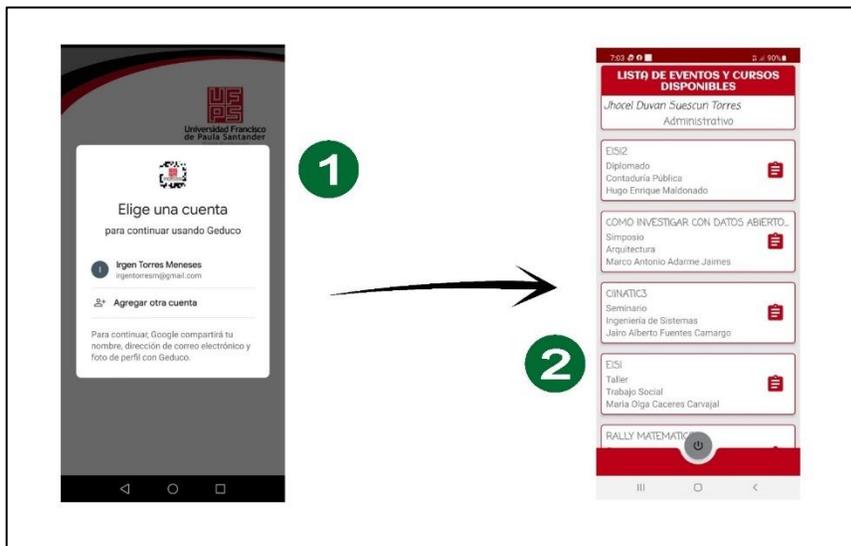


Figura 187 Resultado caso de prueba listar las educaciones continuas en la app

Tabla 93 Caso de prueba login de app móvil de un usuario sin permisos

<b>CASO DE PRUEBA</b>	<b>CP-033</b>
	<b>HU14</b>
	<b>19/04/2021</b>
<b>Nombre</b>	
Listar las jornadas de una educación continua en la app	
<b>Descripción</b>	
El usuario visualiza las jornadas de una educación continua para tomar la asistencia	
<b>Prerrequisitos</b>	
<ul style="list-style-type: none"> <li>• El usuario debe tener rol ROLE_SUPERADMIN, ROLE_MANAECCU o ROLE_ATTENDANCE</li> </ul>	
<b>Resultado esperado</b>	
Visualizar jornadas de una educación continua	
<b>Desarrollo</b>	
<ol style="list-style-type: none"> <li>1. Visualizar listado de educaciones continuas</li> <li>2. Seleccionar una educación continua</li> <li>3. Visualizar las jornadas de la educación continua</li> </ol>	

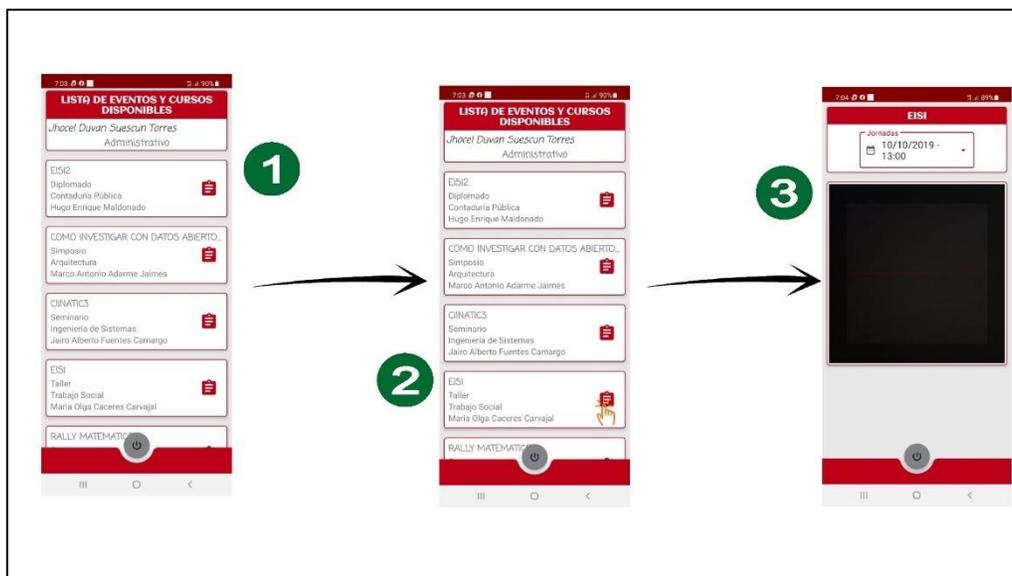


Figura 188 Resultado caso de prueba listar las jornadas de una educación

## ANEXO 10-4. HU-A3 Toma asistencia mediante QR

### Planeación de tareas en tarjeta Kanban

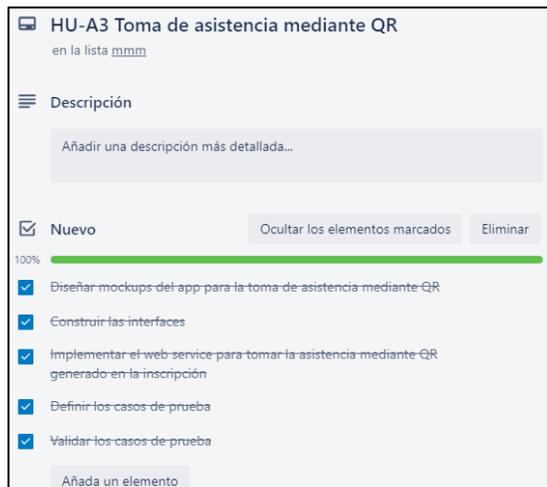


Figura 189 Tarjeta Kanban toma de asistencia mediante QR

### Diseño de mockups

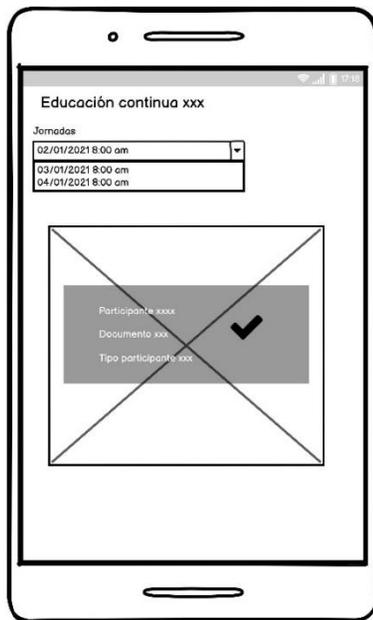


Figura 190 Mockup tomar asistencia mediante

## Definición y validación de casos de pruebas

- Pruebas unitarias

```

@Mock
private IEducacionContinuaService educacionContinuaService;

@Mock
private IPersonaService personaService;

@InjectMocks
private AppRestController appRestController;

private static List<EducacionContinuaAppDto> LIST_EDUCACIONES_CONTINUAS;
private static List<JornadaAppDto> JORNADAS;
private static Map<Integer, ParticipanteDto> MAP_PARTICIPANTE;

@Before
public void init() {
    MockitoAnnotations.initMocks(this);

    LIST_EDUCACIONES_CONTINUAS=new ArrayList<EducacionContinuaAppDto>();

    EducacionContinuaAppDto eduContinua1= new EducacionContinuaAppDto();
    eduContinua1.setId(1L);
    eduContinua1.setNombre("CIINATIC");
    eduContinua1.setFechaInicio(new Date());

    EducacionContinuaAppDto eduContinua2= new EducacionContinuaAppDto();
    eduContinua2.setId(2L);
    eduContinua2.setNombre("EISE");
    eduContinua2.setFechaInicio(new Date());

    LIST_EDUCACIONES_CONTINUAS.add(eduContinua1);
    LIST_EDUCACIONES_CONTINUAS.add(eduContinua2);

    JORNADAS= new ArrayList<JornadaAppDto>();

    JornadaAppDto jornada1= new JornadaAppDto();
    jornada1.setId(1L);
    jornada1.setHoraInicio(new Date());
  
```

Figura 192 Instanciación variables test toma asistencias mediante QR.

```

@Test
public void tomarAsistenciaApp() {
    Mockito.when(educacionContinuaService.tomarAsistencia(1L, 1L,
        "MTUwXzNfNV8xXzEwODQ4NF8xNS0wNC0yMDIwIDA00jA1")).
        thenReturn(MAP_PARTICIPANTE);
    final ResponseEntity<?> response = appRestController.tomarAsistenciaApp(1L, 1L,
        "MTUwXzNfNV8xXzEwODQ4NF8xNS0wNC0yMDIwIDA00jA1");
    assertEquals(response.getStatusCode(), HttpStatus.OK);
}
  
```

Figura 193 Test tomar asistencia app

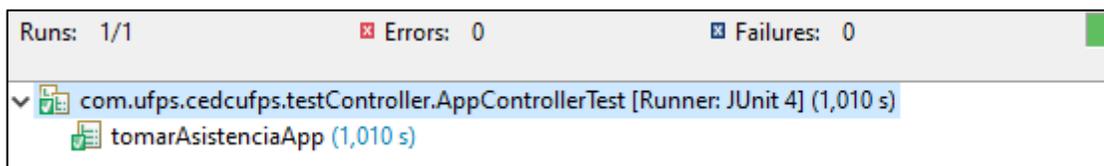


Figura 191 Resultado test tomar asistencia app

- Pruebas de aceptación

Tabla 94 Caso de prueba tomar asistencia a través de la app móvil

CASO DE PRUEBA	CP-034
	HU14
	19/04/2021
<b>Nombre</b> Toma de asistencias a través de la app móvil	
<b>Descripción</b> El usuario toma la lectura del código QR del participante para registrar la asistencia	
<b>Prerrequisitos</b> <ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol ROLE_SUPERADMIN, ROLE_MANAECCU o ROLE_ATTENDANCE</li> </ul>	
<b>Resultado esperado</b> Registro exitoso	
<b>Desarrollo</b> <ol style="list-style-type: none"> <li>1. Seleccionar una educación continua a la cuál se le va a tomar la asistencia</li> <li>2. Seleccionar la jornada en la cuál se va a tomar la asistencia</li> <li>3. Tomar la lectura del código QR del participante</li> <li>4. Visualizar mensaje de registro exitoso</li> </ol>	

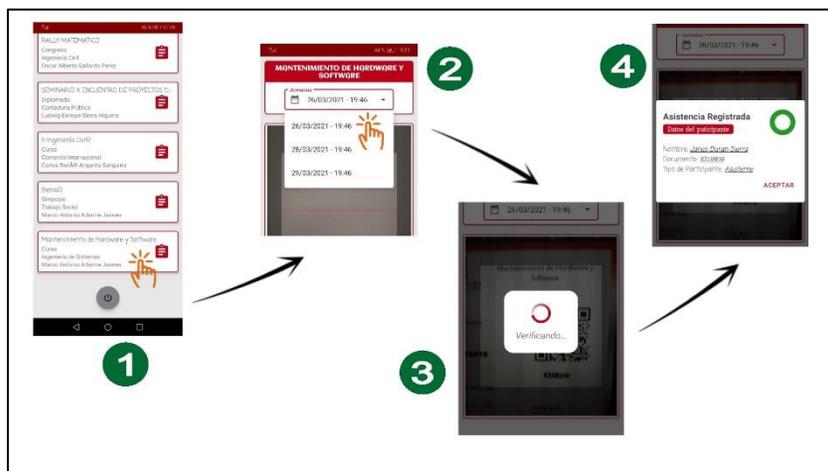


Figura 194 Resultado caso de prueba toma de asistencias a través de la app móvil

Tabla 95 Caso de prueba tomar asistencia de un código QR inválido a través de la app móvil

<b>CASO DE PRUEBA</b>	<b>CP-035</b>
	<b>HU14</b>
	<b>19/04/2021</b>
<b>Nombre</b>	Toma de asistencia de un código QR inválido a través de la app móvil
<b>Descripción</b>	El usuario toma la lectura del código QR que es antiguo o no corresponde a la inscripción del participante
<b>Prerrequisitos</b>	<ul style="list-style-type: none"> <li>• El usuario debe estar logueado</li> <li>• El usuario debe tener rol ROLE_SUPERADMIN, ROLE_MANAECCU o ROLE_ATTENDANCE</li> </ul>
<b>Resultado esperado</b>	Registro fallido
<b>Desarrollo</b>	<ol style="list-style-type: none"> <li>1. Seleccionar una educación continua a la cual se le va a tomar la asistencia</li> <li>2. Seleccionar la jornada en la cual se va a tomar la asistencia</li> <li>3. Tomar la lectura del código QR del participante</li> <li>4. Visualizar mensaje de registro fallido</li> </ol>

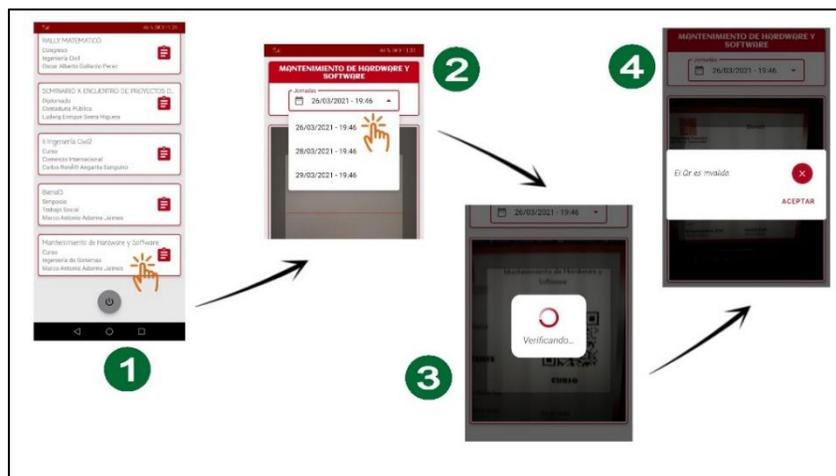


Figura 195 Resultado caso de prueba toma de asistencia de un código QR inválido a través de la app móvil

## ANEXO 11. Scripts de migración

- 1) Se crean los tipos de educación continua que no se tienen registrados en el nuevo sistema

```
INSERT INTO `tipos_educacion_continua`(`id`, `estado_oficial`,
`tipo_educacion_continua`) VALUES (7,0,"Conferencia"), (8,0,"Autoevaluacion");
```

- 2) Se añaden las diferentes modificaciones a los datos para que estos puedan ser migrados con éxito

```
ALTER TABLE personas MODIFY direccion VARCHAR(100);
ALTER TABLE personas MODIFY fecha_expedicion_documento DATETIME
NULL;
ALTER TABLE personas
DROP INDEX UK_email_people;
ALTER TABLE personas MODIFY fecha_nacimiento DATETIME NULL;
ALTER TABLE personas MODIFY id_estado_civil BIGINT NULL;
ALTER TABLE personas MODIFY id_genero BIGINT NULL;
ALTER TABLE educacion_continua MODIFY id_clasificacion_cine BIGINT NULL;
ALTER TABLE educacion_continua MODIFY nombre VARCHAR(125);
```

- 3) Se realiza la inserción de los datos en la tabla personas teniendo en cuenta las transformaciones realizadas

```

INSERT INTO `personas`(`id`,
`primer_nombre`,`segundo_nombre`,`primer_apellido`,`segundo_apellido`,`numero_documento`,`telefono`,`direccion`,`email`,`is_administrativo`,`is_docente`,`is_estudiante`,`is_graduado`,`is_externo`,`id_tipo_documento`,`id_acceso`,`ids_tipo_persona`)
SELECT `id_user`,SUBSTRING_INDEX(`nombres`,``,1) as Primer_Nombre,
IF(SUBSTRING_INDEX(`nombres`,``,-
2)=`nombres`,SUBSTRING_INDEX(`nombres`,``,-
1),SUBSTRING_INDEX(`nombres`,``,-2)) as Segundo_Nombre, `primer_apellido`,
`segundo_apellido`,`documento_identidad`,`telefono`,`direccion`,`email`,
`administrativo`,`docente`,`estudiante_sistemas` or estudiante_otro as "is_estudiante",
`egresado`,`externo`,(case ADEEX.user.id_tipo_documento WHEN 1 THEN 1
WHEN 3 THEN 4 WHEN 4 THEN 3 WHEN 5 THEN 5 end), FLOOR(RAND()*
301028379) + 100 AS `random_number`, id_tp.tper FROM ADEEX.user
JOIN (select group_concat(sq.id_tipo_persona) as tper, sq.id_user as id_per from
(select tp.id as id_tipo_persona, u.id_user from tipos_persona tp, ADEEX.user u where
(tp.id='1' and (u.estudiante_sistemas or u.estudiante_otro)) or (tp.id='2' and u.docente)
or (tp.id='3' and u.administrativo) or (tp.id='4' and u.egresado) or (tp.id='5' and
u.externo) order by u.id_user) sq group by sq.id_user) as id_tp on
id_tp.id_per=ADEEX.user.id_user

```

```
WHERE "random_number" NOT IN (SELECT id_acceso FROM personas);
```

- 4) Se realiza la actualización de los correos cambiando los correos repetidos a un numero aleatorio.

```
UPDATE personas
```

```
SET `email`= FLOOR(RAND() * 3010282) + 100
```

```
WHERE id IN (SELECT id_user FROM ADEEX.user
```

```
WHERE email IN (
```

```
SELECT email FROM ADEEX.user
```

```
GROUP BY email
```

```
HAVING COUNT(*)>1));
```

- 5) Se realiza la inserción de los datos en la tabla educación continua

```
INSERT IGNORE INTO `educacion_continua`(`id`, `cant_max_participantes`,
`duracion`, `estado`, `fecha_inicio`, `fecha_fin`, `fecha_lim_inscripcion`, `info_adicional`,
`nombre`, `id_tipo_educacion_continua`, `lugar`, `id_acceso`, `id_programa`, `id_docente`
)
```

```
SELECT ae.id_evento, `cantidad_max_participantes`, `duracion`,
IF(ae.estado=1, "Terminado", "Terminado") as Estado, `fecha_inicio`, `fecha_fin`,
`fecha_lim_inscripcion`, `resumen`, `nombre`, (case ae.id_tipo_evento WHEN 1
THEN 6 WHEN 2 THEN 7 WHEN 4 THEN 1 WHEN 7 THEN 2 WHEN 8
THEN 1 WHEN 9 THEN 5 WHEN 10 THEN 3 WHEN 11 THEN 4 WHEN 12
THEN 8 end), el.nombre_lugar, FLOOR(RAND() * 301028) + 100 AS
`random_number`, '1', '222222' FROM ADEEX.evento as ae
```

```
JOIN ADEEX.evento_lugarevento as le ON le.id_evento=ae.id_evento
```

```
JOIN ADEEX.lugar_evento as el ON el.id_lugar_evento=le.id_lugar_evento
```

- 6) Se crean los valores del campo consecutivo en la tabla educacion\_continua

```
Update educacion_continua inner join (SELECT id, LPAD(ROW_NUMBER()
OVER(PARTITION BY id_tipo_educacion_continua ORDER BY id ASC),3,0) as
consecutivo FROM educacion_continua) sq
on educacion_continua.id=sq.id
set educacion_continua.consecutivo=sq.consecutivo;
```

- 7) Se hace la inserción de los datos en la tabla participantes

```
INSERT INTO `participantes`(`id`, `aprobado`, `fecha_registro`,
`educacion_continua_id`, `id_persona`, `id_tipo_participante`, `id_tipo_persona`,
`token`) SELECT `id_inscripcion`, `estado`, `fecha_inscripcion`, `id_evento`,
`id_user`,(case ADEEX.inscripcion.id_tipo_paticipante WHEN 3 THEN 1 WHEN 4
THEN 1 WHEN 5 THEN 1 WHEN 6 THEN 1 WHEN 7 THEN 1 WHEN 8 THEN
1 WHEN 9 THEN 2 WHEN 10 THEN 1 WHEN 11 THEN 1 WHEN 12 THEN 1
WHEN 13 THEN 1 end), (case ADEEX.inscripcion.id_tipo_paticipante WHEN 3
THEN 2 WHEN 4 THEN 1 WHEN 6 THEN 3 WHEN 5 THEN 4 WHEN 7
THEN 5 end), FLOOR(RAND() * 301028379) + 100 AS `random_number` FROM
ADEEX.inscripcion WHERE "random_number" NOT IN (SELECT token FROM
participantes);
```

8) Se realiza la inserción de los datos en la tabla jornadas

```
INSERT INTO `jornadas`(`id`, `hora_fin`, `hora_inicio`, `educacion_continua_id`)
SELECT `id_horario`, `hora_inicio`, `hora_fin`, e.id_evento FROM ADEEX.horario
JOIN ADEEX.evento_ponente as ep on ep.id_evento_ponente=
ADEEX.horario.id_evento_ponente
JOIN ADEEX.evento as e on e.id_evento = ep.id_evento and ep.id_evento!= 47;
```

9) Se realiza la inserción de los datos en la tabla Asistencia

```
INSERT INTO `asistencias`(`id`, `id_jornada`, `id_participante`)
SELECT `id_control_asistencia`, `id_horario`, `id_inscripcion` FROM
ADEEX.control_asistencia;
```

10) Se realiza la inserción de los datos en la tabla administrativos

```
INSERT INTO `administrativos`(`id_persona`)
SELECT id FROM personas WHERE is_administrativo=1;
```

11) Se realiza la inserción de los datos en la tabla externos

```
INSERT INTO `externos`(`id_persona`)
SELECT id FROM personas where is_externo=1;
```

12) Se establece nuevamente la clave unica para los emails de las personas que se eliminó en el punto número 2

```
ALTER TABLE personas
ADD CONSTRAINT UK_email_people UNIQUE (email);
```

13) Se agregan los permisos a los usuarios para acceder a la aplicación web

```
INSERT IGNORE INTO `personas_x_roles`(`id_rol`, `id_persona`)
SELECT 5, id FROM personas;
```

14) Se crea la estructura de los diplomas de las educaciones continuas existentes

```
Insert into diplomas (id, structure_diploma, created_by, created_at, updated_by, updated_at)
Select e.id, , 'system', (select now()), 'system', (select now())
From educacion_continua e

Update educacion_continua e join diplomas d on e.id=d.id set id_diploma=d.id
```