



Universidad del Azuay
Facultad de Ciencias de la Administración
Escuela de Ingeniería de Sistemas y Telemática
“Aplicativos web y móvil para el Test de Margaritas”

Tesis previa a la obtención del Título de Ingeniero de Sistemas
y Telemática

Autores:

Jennifer Cecibel Torres Hurtado

Jonnathan José Zhunio Ochoa

Directora: Ing. Catalina Astudillo

Cuenca, Ecuador

2014

DEDICATORIA

Este trabajo de tesis de grado está dedicada a nuestros padres: Juan, Rosa, José y Fanny, ya que nos enseñaron desde pequeños a luchar para alcanzar nuestros sueños, contando siempre con su apoyo y brindándonos las fuerzas necesarias para llegar a esta instancia de nuestros estudios. Nos han dado los valores, principios, carácter, perseverancia y coraje para conseguir nuestros objetivos.

A nuestros hermanos, por estar siempre presentes en los buenos y malos momentos de la vida.

A nuestros demás familiares y amigos, quienes en momentos difíciles supieron brindarnos una sonrisa, un abrazo o un consejo, motivándonos para seguir adelante.

AGRADECIMIENTOS

Nos complace exteriorizar nuestros sinceros agradecimientos a la Universidad del Azuay, Facultad de Ciencias de la Administración, Escuela de Ingeniería en Sistemas y Telemática, y en ellas a los distinguidos docentes quienes con su profesionalismo y ética supieron brindarnos sus conocimientos y con ellos forjamos nuestras bases para convertirnos en futuros profesiones, útiles para la sociedad.

A nuestra Directora Ingeniera Catalina Astudillo, quien nos ha brindado su experiencia, tiempo y el conocimiento necesario para culminar con éxitos esta tesis. Gracias por ser nuestra guía durante este proceso.

ÍNDICE DE CONTENIDOS

DEDICATORIA	ii
AGRADECIMIENTOS	iii
ÍNDICE DE CONTENIDOS	iv
ÍNDICE DE ILUSTRACIONES.....	viii
ÍNDICE DE TABLAS	xiv
ÍNDICE DE ANEXOS.....	xv
RESUMEN.....	xvi
ABSTRACT.....	xvii
INTRODUCCIÓN	xviii
CAPÍTULO 1: TEST DE MARGARITAS.....	1
1.1. Pruebas Psicométricas de Habilidades Específicas	1
1.2. Caras – Test de Percepción de Diferencias (TPD- C).....	1
1.3. Justificación Estadística	3
1.3.1. Fiabilidad	3
1.4. Caras – R. Media de atención sostenida y del control de impulsividad en niños. 4	
1.4.1. Aciertos (A).-	5
1.4.2. Errores (E).-.....	5
1.4.3. Aciertos netos (A – E).-	5
1.4.4. Índice de Control de la Impulsividad (ICI).-.....	6
1.5. Aplicaciones del Test de Margaritas	6
1.6. Investigación exploratoria	7
CAPÍTULO 2: MODELO DE ANÁLISIS	8
2.1. Requisitos Funcionales.....	8
2.2. Requisitos No Funcionales	9
2.2.1. Usabilidad.-	9

2.2.2.	Funcionalidad.-.....	10
2.2.3.	Simplicidad.-	10
2.2.4.	Flexibilidad.-	10
2.2.5.	Portabilidad.-	10
2.2.6.	Confiabilidad.-	10
2.3.	Referencias	10
2.4.	Diagrama de Actores	13
2.5.	Diagrama de Casos de Uso.....	14
2.6.	Especificación Casos de Uso.....	15
2.7.	Casos de Uso	15
2.8.	Diagrama de Clases	18
2.9.	Diagrama de Secuencia	19
2.9.1.	Diagrama de Secuencia – Mantenimiento Test.....	19
2.9.2.	Diagrama de Secuencia – Generar Ficha	20
2.9.4.	Diagrama de Secuencia – Tomar Test	22
2.9.5.	Diagrama de Secuencia – Extraer Información	23
CAPÍTULO 3: IMPLEMENTACIÓN.....		24
3.1.	Desarrollo Base de Datos	25
3.1.1.	Definición de MySQL.....	25
3.1.2.	Explicación de MySQL.....	25
3.1.3.	Definición de SQLite	26
3.1.4.	Explicación de SQLite	26
3.2.	Desarrollo Android.....	27
3.2.1.	Definición del Sistema Operativo Android.....	27
3.2.2.	Arquitectura del Sistema Operativo Android.....	27
3.2.3.	Estructura de un Proyecto Android	29
3.2.4.	Explicación de Android.....	32

3.3.	Desarrollo iOS	33
3.3.1.	Definición del Sistema Operativo iOS	33
3.3.2.	Arquitectura del Sistema Operativo iOS	33
3.3.3.	Explicación de iOS	34
3.4.	Desarrollo Web	35
3.4.1.	Conceptos Utilizados	35
3.4.2.	Explicación de la parte Web	36
3.5.	Manuales de Usuario – Web	39
3.5.1.	Iniciar Sesión	39
3.5.2.	Menú Usuario	41
3.5.3.	Menú Procesos	44
3.5.4.	Menú Tomar Test	58
3.5.5.	Menú Acerca de	63
3.5.6.	Menú Administración	63
	Manuales de Usuario – Android e iOS	84
3.5.7.	Descargar aplicación Android	84
3.5.9.	Iniciar Sesión	87
3.5.10.	Menú Acerca de	91
3.5.11.	Menú Inicio	92
3.5.12.	Menú Fichas	93
3.5.13.	Menú Tomar Test Android	99
3.5.14.	Menú Tomar Test en iOS	100
3.5.15.	Menú Sincronizar	103
	CAPÍTULO 4: PRUEBAS	107
4.1.	Página Web	107
4.2.	Aplicación Android	108
	CONCLUSIONES	110

RECOMENDACIONES	111
GLOSARIO	112
BIBLIOGRAFÍA	115
ANEXOS	119

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Diagrama de Actores.....	13
Ilustración 2: Diagrama de Casos de Uso	14
Ilustración 3: Diagrama de Clases.....	18
Ilustración 4: Diagrama de Secuencia – Mantenimiento Test	19
Ilustración 5: Diagrama de Secuencia – Generar Ficha	20
Ilustración 6: Diagrama de Secuencia – Agendar Citas.....	21
Ilustración 7: Diagrama de Secuencia – Tomar Test	22
Ilustración 8: Diagrama de Secuencia – Extraer Información	23
Ilustración 9: Arquitectura Test de Margaritas	24
Ilustración 10: Capas Android	28
Ilustración 11: Estructura del Test de Margaritas, visualizado desde Eclipse	31
Ilustración 12: Arquitectura Test de Margaritas - iOS.....	34
Ilustración 13: Arquitectura Test de Margaritas - Página Web.....	36
Ilustración 14: Página Web – inicio de sesión	39
Ilustración 15: Página Web – captcha correcto.....	40
Ilustración 16: Página Web – mensaje: Nombre de usuario o contraseña incorrectos	40
Ilustración 17: Página Web – pantalla de bienvenida, menú principal	41
Ilustración 18: Página Web – submenú usuario.....	41
Ilustración 19: Página Web – modificación de los datos de perfil.....	42
Ilustración 20: Página Web – mensaje: El usuario se ha modificado correctamente. 42	
Ilustración 21: Página Web – cambio de contraseña	43
Ilustración 22: Página Web – mensajes: La contraseña actual es incorrecta, La nueva contraseña debe tener la menos 6 caracteres.....	43
Ilustración 23: Página Web – mensaje: La contraseña se ha modificado correctamente	44
Ilustración 24: Página Web – al seleccionar el botón cerrar sesión, el sistema vuelve a la pantalla de inicio de sesión.....	44
Ilustración 25: Página Web – submenú de procesos	45
Ilustración 26: Página Web – pantalla de citas programadas.....	45
Ilustración 27: Página Web – ingreso de una nueva cita	46
Ilustración 28: Página Web – mensaje: ingreso correcto de la cita.....	46

Ilustración 29: Página Web – selección de la cita a editar	46
Ilustración 30: Página Web – modificación de datos de la cita	47
Ilustración 31: Página Web – mensaje: modificación correcta de la cita	47
Ilustración 32: Página Web – vista de impresión de citas.....	48
Ilustración 33: Página Web – selección la cita a anular.....	48
Ilustración 34: Página Web – mensaje de confirmación	49
Ilustración 35: Página Web – mensaje: anulación correcta de la cita	49
Ilustración 36: Página Web – pantalla de fichas ingresadas	49
Ilustración 37: Página Web – ingreso de una nueva ficha	50
Ilustración 38: Página Web – mensaje: ingreso correcto de la ficha	50
Ilustración 39: Página Web – lista de todas las fichas ingresadas	50
Ilustración 40: Página Web – selección de la ficha a editar, primera opción	51
Ilustración 41: Página Web – selección de la ficha a editar, segunda opción.....	51
Ilustración 42: Página Web –modificación de los datos de la ficha	51
Ilustración 43: Página Web – mensaje: ingreso correcto de la entrevista.....	52
Ilustración 44: Página Web – selección de la ficha que se va a entrevistar, primera opción.....	52
Ilustración 45: Página Web – selección de la ficha que se va a entrevistar, segunda opción.....	52
Ilustración 46: Página Web – ingreso de una nueva entrevista.....	53
Ilustración 47: Página Web – vista de impresión de fichas	53
Ilustración 48: Página Web – selección del botón buscar.....	54
Ilustración 49: Página Web – ingreso de caracteres a buscar	54
Ilustración 50: Página Web – selección de la ficha a anular.....	54
Ilustración 51: Página Web – mensaje de confirmación.....	55
Ilustración 52: Página Web – mensaje: anulación correcta de la ficha.....	55
Ilustración 53: Página Web – lista de todas las fichas existentes	55
Ilustración 54: Página Web – lista de todas las fichas existentes, selección de resultados	56
Ilustración 55: Página Web – lista de todos los test tomados a la ficha seleccionada	56
Ilustración 56: Página Web – vista de impresión de los test tomados	57
Ilustración 57: Página Web – detalle del test seleccionado	57
Ilustración 58: Página Web – vista de impresión del test seleccionado.....	58
Ilustración 59: Página Web – selección de resultados agrupar por expresión	58

Ilustración 60: Página Web – menú tomar test	58
Ilustración 61: Página Web – lista de todas las fichas existentes, ver ficha	59
Ilustración 62: Página Web – vista de la información de la ficha seleccionada	59
Ilustración 63: Página Web – selección de la ficha a tomar el test	60
Ilustración 64: Página Web – lista de todos los test existentes	60
Ilustración 65: Página Web – ingreso de caracteres a buscar	61
Ilustración 66: Página Web – selección del test que se va a evaluar	62
Ilustración 67: Página Web – vista del test seleccionado	62
Ilustración 68: Página Web – mensaje de información.....	63
Ilustración 69: Página Web – mensaje: ingreso correcto del test.....	63
Ilustración 70: Página Web – vista del menú acerca de.....	63
Ilustración 71: Página Web – submenú administración.....	64
Ilustración 72: Página Web – pantalla de motivos consulta existentes.....	64
Ilustración 73: Página Web – ingreso de un nuevo motivo consulta.....	64
Ilustración 74: Página Web – mensaje: ingreso correcto del motivo consulta	65
Ilustración 75: Página Web – lista de todos los motivos consulta ingresados	65
Ilustración 76: Página Web – selección del motivo consulta a editar.....	65
Ilustración 77: Página Web – modificación de los datos del motivo consulta.....	66
Ilustración 78: Página Web – mensaje: modificación correcta del motivo consulta .	66
Ilustración 79: Página Web – vista de impresión de motivo consulta	66
Ilustración 80: Página Web –ingreso de caracteres a buscar	67
Ilustración 81: Página Web – selección del motivo consulta a anular.....	67
Ilustración 82: Página Web – mensaje de confirmación.....	67
Ilustración 83: Página Web – mensaje: anulación correcta del motivo consulta.....	68
Ilustración 84: Página Web – lista de todos los motivos consulta existentes	68
Ilustración 85: Página Web – pantalla de test ingresados	68
Ilustración 86: Página Web – ingreso de un nuevo test – datos generales.....	69
Ilustración 87: Página Web – ingreso de un nuevo test – selección de margaritas....	69
Ilustración 88: Página Web – ingreso de un nuevo test – cambio del orden de las margaritas.....	70
Ilustración 89: Página Web – ingreso de un nuevo test – guardar las margaritas seleccionadas.....	70
Ilustración 90: Página Web – mensaje: ingreso correcto del test.....	71
Ilustración 91: Página Web – lista de todos los test ingresados.....	71

Ilustración 92: Página Web – selección del test a editar	71
Ilustración 93: Página Web – modificación de datos del test	72
Ilustración 94: Página Web – mensaje: modificación correcta del test	72
Ilustración 95: Página Web – Vista de la impresión de los test	73
Ilustración 96: Página Web – selección e ingreso de los caracteres a buscar	73
Ilustración 97: Página Web – selección del test a anular	74
Ilustración 98: Página Web – mensaje de confirmación	74
Ilustración 99: Página Web – mensaje: anulación correcta del test	74
Ilustración 100: Página Web – lista de todos los test anulados	75
Ilustración 101: Página Web – pantalla de usuarios existentes.....	75
Ilustración 102: Página Web – ingreso de un nuevo usuario	75
Ilustración 103: Página Web – mensaje: cédula ya existente	76
Ilustración 104: Página Web – mensaje: usuario ya existente	76
Ilustración 105: Página Web – mensaje: ingreso correcto del usuario	76
Ilustración 106: Página Web – lista de todos los test ingresados.....	76
Ilustración 107: Página Web – selección del usuario a editar.....	77
Ilustración 108: Página Web – Modificación de los datos del usuario	77
Ilustración 109: Página Web – mensaje: modificación correcta del usuario	78
Ilustración 110: Página Web – selección e ingreso de los caracteres a buscar	78
Ilustración 111: Página Web – selección del usuario a anular.....	78
Ilustración 112: Página Web – mensaje de confirmación	79
Ilustración 113: Página Web – mensaje: anulación correcta del usuario.....	79
Ilustración 114: Página Web – lista de todos los usuarios existentes	79
Ilustración 115: Página Web – lista de todos los usuarios anulados.....	79
Ilustración 116: Página Web – pantalla de escuelas ingresadas	80
Ilustración 117: Página Web –ingreso de una nueva escuela	80
Ilustración 118: Página Web – mensaje: ingreso correcta de la escuela.....	80
Ilustración 119: Página Web – lista de todas las escuelas ingresadas.....	81
Ilustración 120: Página Web – selección de la escuela a anular	81
Ilustración 121: Página Web – mensaje de confirmación	81
Ilustración 122: Página Web – mensaje: anulación correcta de la escuela.....	81
Ilustración 123: Página Web – pantalla de selección de imágenes.....	82
Ilustración 124: Página Web – selección de la expresión e imagen a ingresar.....	82
Ilustración 125: Página Web – almacenamiento de las imágenes subidas.....	83

Ilustración 126: Android – play store – ícono de la tienda virtual de android.....	84
Ilustración 127: Android – play store – búsqueda de la aplicación Test de Margaritas	85
Ilustración 128: Android – play store – selección del botón de instalación	85
Ilustración 129: Android – play store – permisos e inicio de la instalación	86
Ilustración 130: Android – play store – instalación completa, ícono de la aplicación Test de Margaritas.....	87
Ilustración 131: Android – inicio de sesión	88
Ilustración 132: iOS - inicio sesión.....	88
Ilustración 133: Android - mensaje: La primera vez que inicia sesión, debe tener conexión a internet	89
Ilustración 134: Android - mensaje: Usuario/Contraseña son incorrectos.....	89
Ilustración 135: iOS - mensaje: Inicio de sesión incorrecto	90
Ilustración 136: Android - iniciando sesión	90
Ilustración 137: iOS - pantalla de inicio	91
Ilustración 138: Android - pantalla de inicio	91
Ilustración 139: Android - submenú acerca de	92
Ilustración 140: Android - pantalla acerca de	92
Ilustración 141: Android - pantalla de bienvenida.....	93
Ilustración 142: Android - menú principal.....	93
Ilustración 143: Android – mensaje: No existen datos disponibles	94
Ilustración 144: Android - lista de todas las fichas ingresadas	94
Ilustración 145: Android - ingreso de caracteres a buscar	95
Ilustración 146: Android - selección de la ficha a editar	95
Ilustración 147: Android - modificación de datos de la ficha.....	96
Ilustración 148: Android - mensaje: datos guardados correctamente	96
Ilustración 149: Android - ingresar nueva ficha	97
Ilustración 150: Android - ingreso de una nueva ficha	97
Ilustración 151: Android - mensaje: datos guardados correctamente	98
Ilustración 152: Android - mensaje confirmación	98
Ilustración 153: Android - selección del test.....	99
Ilustración 154: Android - selección del paciente.....	99
Ilustración 155: Android - instrucciones y toma del test	100
Ilustración 156: Android - mensaje confirmación	100

Ilustración 157: iOS - lista de todos los pacientes	101
Ilustración 158: iOS - buscar pacientes.....	101
Ilustración 159: iOS - botón continuar.....	102
Ilustración 160: iOS - seleccionar el test.....	102
Ilustración 161: iOS - mostrar el test	103
Ilustración 162: iOS - mensaje de confirmación.....	103
Ilustración 163: Android - menú sincronizar	104
Ilustración 164: Android - mensaje confirmación datos actualizados	104
Ilustración 165: Android - mensaje confirmación información subida	105
Ilustración 166: Android - descargar fichas por escuela.....	105
Ilustración 167: Android - mensaje confirmación descarga correcta.....	106
Ilustración 168: Android - mensaje confirmación vaciar fichas	106

ÍNDICE DE TABLAS

Tabla 1: Requerimiento de función de ingreso	11
Tabla 2: Requerimientos de función toma del test.....	11
Tabla 3: Requerimientos de función de resultados	12
Tabla 4: Requerimientos de función de manejo de resultados.....	12
Tabla 5: Actor Administrador	15
Tabla 6: Actor Investigador	15
Tabla 7: Actor Sujeto Evaluación	15
Tabla 8: Mantenimiento Test	16
Tabla 9: Generar Ficha.....	16
Tabla 10: Agendar Citas.....	16
Tabla 11: Tomar Test	17
Tabla 12: Extraer Información	17
Tabla 13: Tabla funcionamiento navegadores	108
Tabla 14: Tabla funcionamiento aplicación Android.....	109

ÍNDICE DE ANEXOS

1. Cómo instalar Android en Windows.
2. Estructura de la Base de Datos en MySQL utilizado en el Test.
3. Triggers utilizados en el Test de Margaritas.
4. Fotos pruebas con niños.

RESUMEN

El propósito de esta tesis es sistematizar el Test de Margaritas, para evaluar diversos procesos de atención en niños de cinco a doce años de edad, mediante la utilización de software basado en dispositivos táctiles. Este test está desarrollado para plataformas: web, Android e iOS; en el entorno web la implementación se orienta al sistema de gestión administrativa, y en el entorno móvil del sistema para la aplicación del test a los niños. La finalidad es obtener los resultados de cada test realizado, es decir aciertos, equivocaciones, tiempos, etc., y agruparlos para su posterior análisis.

ABSTRACT

The purpose of this thesis is to systematize the Test of Margaritas to evaluate various attention processes in five to twelve years old children, by using software based on touch devices. This test is developed for Web, Android and iOS platforms. In the web environment the implementation aims at the administrative management system; and in the system's mobile environment at the application of the children's test. The purpose is to obtain the results of each test performed, in other words, achievements, errors, times, etc. and group them for further analysis.



A handwritten signature in blue ink, appearing to read 'Lourdes Crespo'.

Translated by,
Lic. Lourdes Crespo

INTRODUCCIÓN

Este trabajo se enfoca en el desarrollo de un aplicativo web y móvil para medir la atención sostenida y selectiva de un niño o niña, a través de la utilización de un test llamado “Test de Margaritas”. Mediante el uso de nuevas tecnologías y la contribución de la Escuela de Psicología Clínica de la Universidad del Azuay, se pretende recoger datos confiables para su posterior análisis.

Este documento se divide en cuatro partes:

- En la primera parte se detallan los conceptos utilizados en el test, se analiza el test de percepción de diferencias-caras (TPD-C), las medidas de atención y control de impulsividad, estos son la base para la creación del test; además se muestra las posibles aplicaciones del test y se presenta una investigación exploratoria.
- En la segunda parte se analizan los requisitos funcionales y no funcionales de la aplicación en conjunto con los diagramas de actores, diagrama de casos de uso y diagrama de secuencia.
- En la tercera parte se explican conceptos utilizados en la aplicación (MySQL, SQLite, PHP, etc.) y se presentan los manuales de usuario de cada aplicación (Android, iOS y página Web) junto con gráficos y su debida explicación.
- Por último, se presentan tablas comparativas de la compatibilidad de las aplicaciones en diferentes entornos, además se detalla los métodos de seguridad utilizados en Android y la página Web.

De esta forma estamos contribuyendo con la Institución Superior que nos permitió una formación acorde con las exigencias de la pedagogía actual, una pedagogía constructivista, que apunta al desarrollo de aprendizajes a partir de actividades concretas, desarrolladas en escenarios reales, en este caso un test que será utilizado para resolver problemas reales en un mundo real.

CAPÍTULO 1: TEST DE MARGARITAS

La Escuela de Ingeniería de Sistemas conjuntamente con la Escuela de Psicología Clínica desarrolló un proyecto de investigación para trabajar en la creación de una prueba de atención, que finalmente se denominó, “Test de Margaritas”.

Para comprender el funcionamiento del Test de Margaritas se debe investigar el Test de Caras presentado por Thurstone & Yela, ya que la lógica de ambos es la misma. Para evitar quebrantar derechos de autor, se realizó una variación en el nombre de test y las imágenes utilizadas; la principal diferencia está en las imágenes, ya que en el Test de Caras se utiliza boca, ojos, cejas y pelo, mientras que el Test de Margaritas utiliza boca, ojos, cejas y pétalos. (Thurstone & Yela, Percepción de Diferencias (caras) Manual., 2009)

1.1. Pruebas Psicométricas de Habilidades Específicas

Es la medición objetiva y estandarizada de una muestra de comportamiento humano, se somete a diferentes exámenes bajo condiciones reguladas. Este tipo de prueba busca medir dos conceptos básicos: aptitud y aprovechamiento.

Aptitud.- son las distintas capacidades que una persona determinada tiene para realizar adecuadamente una tarea.

Aprovechamiento.- es la obtención de algún beneficio, generalmente vinculado con el desarrollo de alguna actividad, negocio o empleo.

1.2. Caras – Test de Percepción de Diferencias (TPD- C)

El TPD-C, cuyos autores son Thurstone y Yela, es un test que evalúa la atención y las aptitudes perceptivo-imaginativas, que se apoyan en principios de semejanza – diferencias. Pertenece al tipo de test que mide las habilidades específicas correspondientes al “Aprovechamiento” de una habilidad determinada. (Thurstone & Yela, Percepción de Diferencias (caras) Manual., 2009)

El TPD-C es una matriz de estímulos dados que arroja un nivel de aprovechamiento de dicha habilidad, es decir, son pruebas de discriminación que constan de sesenta elementos gráficos y ponen en juego la habilidad visual para encontrar un parecido, igualdad o diferencia, a lo largo de una serie de tres imágenes esquemáticas de rostros con boca, ojos, cejas y pelo representados con trazo elementales básicamente iguales. De las tres caras representadas de forma horizontal dos son iguales y una diferente, la tarea consiste en identificar la cara diferente y tacharla. (Thurstone & Yela, Caras-R Test de Percepción de Diferencias - Revisado., 2012)

Según (James, 1980): *“La atención es el proceso por el que la mente toma posesión, de forma vívida y clara, de uno de los diversos objetivos o trenes de pensamiento que aparecen simultáneamente. Focalización y concentración de la conciencia son su esencia. Implica la retirada del pensamiento de varias cosas para tratar efectivamente otras”*.

Contrariamente al optimismo de James, y de acuerdo con (Dark & Johnston, 1982): *“no existe una definición clara y universal del concepto de atención. Existen tres dificultades básicas a la hora de acotar los estudios sobre ésta. Tales dificultades corresponde a: 1) su definición conceptual divergente basándose en los diferentes fenómenos que engloba, 2) la multiplicidad de teorías que pueden dar cuenta de unos mismos datos empíricos, y 3) la apelación frecuente a metáforas frente a la imposibilidad de una definición conceptual simple”*.

En el Test de Percepción de Diferencias, la atención no es unitaria, se puede diferenciar entre:

- **Atención sostenida o vigilancia.**- según (Parasuraman, 1984): *“La habilidad para mantener la atención y permanecer vigilantes para tales eventos durante periodos sostenidos de tiempo, forma el foco de la investigación sobre la atención sostenida o vigilancia”*. Como ejemplos tenemos conducir durante la noche, probar algún juego en el Kinect o practicar algún deporte extremo.

- **Atención selectiva.-** según (Johnston & Dark, 1986): “*La atención selectiva o focalizada, es la capacidad para atender a uno o dos estímulos relevantes sin confundirse ante el resto de estímulos que actúan como distractores*”. Es decir, prioriza los estímulos importantes para la persona, sin considerar el resto de estímulos que lo acompañan. Como ejemplo, se tiende a atender a una persona que habla en un ambiente ruidoso.
- **Atención dividida o control.-** Según (Kahneman, 1973): “*La atención dividida es la capacidad para responder, al menos, a dos tareas al mismo tiempo*”. Generalmente estos procesos suelen estar relacionados con actividades rutinarias y mecanizadas. Como el siguiente ejemplo que se refiere a la acción de conducir, planteado por (García, 1997): “*...mientras vamos conduciendo hablamos con nuestro acompañante, miramos por el espejo retrovisor para comprobar si alguien quiere adelantarnos, tenemos que controlar el movimiento de los pedales y la palanca de marchas, etc.*”

1.3. Justificación Estadística

1.3.1. Fiabilidad

El coeficiente de fiabilidad indica la exactitud o estabilidad estadística de los resultados. Muestra un porcentaje de errores causales en las pruebas; por ejemplo, si la fiabilidad de las pruebas es del 0.90, quiere decir que: del 100% de la muestra el 90% es una medida auténtica, mientras que, el 10% son errores aleatorios. (Thurstone & Yela, Caras-R Test de Percepción de Diferencias - Revisado., 2012)

Esta prueba se distribuye principalmente entre los factores de rapidez de percepción (P) e inteligencia espacial (S).

- **Rapidez de Percepción.-** según (Fernández Muñoz, 2011): “*La habilidad para identificar elementos semejantes, así como para discriminar elementos diferentes*”.

- **Inteligencia espacial.-** según (Gardner, 1983): *“Es la capacidad de pensar en tres dimensiones. Permite percibir imágenes externas e internas, recrearlas, transformarlas o modificarlas, recorrer el espacio o hacer que los objetos lo recorran y producir o decodificar información gráfica”*.

1.4. Caras – R. Media de atención sostenida y del control de impulsividad en niños.

Los autores del libro medida de atención sostenida y del control de la impulsividad en niños: nueva modalidad de aplicación del Test de Percepción de Diferencias “Caras”, (Crespo, Eguílaz, Narbona, Perlta, & Reparaz, 2006) estudiaron la utilidad del TPD-C como medida de atención sostenida y del control de respuestas impulsivas en niños y niñas (a esta prueba se la llama CARAS – R). Dada la fácil aplicación de esta prueba y al mismo tiempo, lo arduo y tedioso que puede ser la misma en los primeros minutos, el equipo de Crespo-Eguílaz y cols., añadió al tiempo de aplicación original – tres minutos – un segundo periodo de tres minutos extras, para alargar el tiempo de aplicación a seis minutos. Los resultados de dicho estudio evidenciaron que la aplicación de esta prueba hacía a los individuos sensibles a la fatiga atencional, ya que el rendimiento descendía significativa en el segundo periodo, con respecto al primero.

Se debe recalcar que en este tipo de prueba, el sujeto que responde a treinta ítems correctos sin ningún erróneo, no presenta las mismas características ante el sujeto que responde treinta ítems correctos con diez erróneos. Ante el mismo número de aciertos, el primer sujeto muestra un mayor grado de eficiencia ya que no ha cometido ningún error.

A partir del número de respuestas correctas y número de respuestas erróneas, calcularon un índice de control de impulsividad (ICI) y aportaron los estadísticos normales de este índice para cada grupo. El equipo en cuestión, aplicó esta prueba a grupos con trastorno de déficit de atención e hiperactividad y a otro grupo con trastorno de aprendizaje de la lectura.

La prueba exige una toma de decisiones por comparación igual – diferente, además implica un considerable esfuerzo de atención sostenida y un buen control para evitar las respuestas reiterativas o al azar. Razón por la cual, esta prueba se puede considerar una herramienta para la evaluación de las capacidades de control atencional y un instrumento útil al momento de evaluar los trastornos por déficit de atención e hiperactividad – impulsividad.

De esta forma la implementación de CARAS – R se basará en las siguientes medidas:

- 1.4.1. Aciertos (A).**- Número total de aciertos. Se concederá un punto por cada una de las respuestas correctas, la suma de estos puntos se consideran como número de aciertos.
- 1.4.2. Errores (E).**- Número total de errores. Se contará el número de respuestas incorrectas y el número de respuestas en blanco, sumando las repuestas incorrectas más los espacios en blanco, se obtendrá el número de errores.
- 1.4.3. Aciertos netos (A – E).**- en Caras – R se considera como medida principal el rendimiento del sujeto, ya que esta medida corregida, mide la eficacia real del sujeto penalizando los errores y considerando omisiones y comisiones.

Se calcula restando el número total de aciertos (A) del número total de errores (E). De esta forma obtenemos una medida más precisa y la eficacia de las respuestas durante la prueba.

- Puntuaciones altas, indicarán que el sujeto procesa de forma rápida los detalles de estímulos visuales y es preciso en los juicios que realiza.
- Puntuaciones medias, reflejan una adecuada capacidad viso perceptiva y atencional.
- Puntuaciones bajas, mostrarán que el sujeto tiene una baja capacidad viso perceptiva y atencional. No realiza juicios de semejanza–diferencia adecuados y no presta la suficiente atención a los detalles.

1.4.4. Índice de Control de la Impulsividad (ICI).- Aporta información sobre si el patrón de respuestas del sujeto es impulsivo o no, y permite componer deducciones sobre su rendimiento. Este índice expresa el nivel de control de la impulsividad del sujeto a la hora de ejecutar cierta tarea, reflejando su estilo cognitivo.

Se calcula dividiendo el número de Aciertos Netos ($A - E$) entre el número de respuestas dadas por el sujeto ($A + E$). Para facilitar la interpretación se optó por multiplicar el índice por 100, para eliminar cifras decimales. Así la fórmula para su cálculo sería: $[(A-E)/(A+E)) * 100]$.

- Las puntuaciones cercanas a 100, indicarán que el sujeto tiene un adecuado control de la impulsividad, ya que ejecutó de forma reflexiva la tarea y, por tanto, cometió pocos errores.
- Las puntuaciones bajas mostrarán que el sujeto es impulsivo en la ejecución de la tarea, no es reflexivo a la hora de realizar los juicios de semejanza – diferencia.

Las puntuaciones en el ICI pueden ser analizadas de forma independiente o en conjunto con la medida de rendimiento ($A - E$), de forma que se puede extraer información sobre el estilo de respuesta del sujeto.

1.5. Aplicaciones del Test de Margaritas

El test de Margaritas es aplicable en niños, desde los cinco hasta los doce años de edad, a los que se pretende medir la atención sostenida y selectiva.

Se requiere de poca formación cultural para realizar el test, de esta manera se asegura que cualquier individuo tenga la capacidad de comprender las instrucciones que debe seguir.

En un principio, el test se aplicará en la población escolar del cantón Cuenca - Ecuador, para esto se evaluará a un número significativo de estudiantes escolares, tanto en el área urbana y rural.

A futuro se tiene planeado aumentar el número de escolares evaluados, se planea difundir la aplicación del test a nivel país, la expansión dependerá del éxito obtenido en el cantón Cuenca.

1.6. Investigación exploratoria

La Ingeniera Patricia Ortega, docente de la Universidad del Azuay, junto con estudiantes de la Escuela de Ingeniería de Sistemas y Telemática (en calidad de pasantes), desarrollaron con anterioridad una aplicación de escritorio y Kinect, la cual permite sistematizar la recogida de datos para este test.

La implementación de Kinect no tuvo la aceptación que se esperaba, principalmente porque el tiempo entre la selección y la detección del Kinect interfería en la medida de la atención.

Cuando los niños realizaron el test en la aplicación de escritorio mostraron menor interés y el test se limitaba únicamente a los niños que sabían cómo manejar el ratón de la computadora.

La implementación del sistema de escritorio tuvo limitantes, ya que cuando se quería ampliar el número de usuarios necesariamente se tenía que acudir a instalar el software en las computadoras destinadas a realizar el test, esto limita el libre acceso al sistema. Además existe el riesgo de que la aplicación sea plagiada, ya que está instalada localmente en cada máquina. (Ortega Chasi, 2012)

Por otro lado, hemos realizado una búsqueda de software desarrollado por otras personas o instituciones para la toma del Test de Caras o el Test de Margaritas en la web y tiendas virtuales de aplicaciones móviles, pero no hemos encontrado aplicaciones destinadas para sistematizar estos test. Debemos aclarar que no podemos asegurar que no exista otro(s) software, pero al menos en nuestra indagación no encontramos tal(es).

CAPÍTULO 2: MODELO DE ANÁLISIS

Los diagramas presentes en este capítulo fueron construidos de acuerdo a las especificaciones de UML indicadas por: (Alejandro Ramirez, 2011).

2.1. Requisitos Funcionales

- 2.1.1. Se podrá medir de manera independiente los aspectos de los procesos de atención de acuerdo a las necesidades determinadas para el proceso de investigación.
- 2.1.2. Cada usuario tendrá un código único, este código se generará de acuerdo a los privilegios dados previamente.
- 2.1.3. El sistema debe permitir el almacenamiento de los resultados, lo que permitirá la generación de informes que muestren los resultados obtenidos por el sujeto evaluación.
- 2.1.4. En el sistema se establecen dos partes, la primera consta sobre la administración y mantenimiento del sistema, y la segunda parte sobre la selección e ingreso del test.
- 2.1.5. Para la administración del sistema intervienen los usuarios administrador e investigador; el usuario administrador es el único que puede realizar mantenimientos, mientras que el usuario investigador solamente puede extraer información ingresada previamente, pero este no puede modificar ni eliminar dicha información.
- 2.1.6. En el ingreso del test intervienen tres tipos de usuarios que son administrador, investigador y sujeto evaluación; tanto el administrador como el investigador pueden escoger al sujeto evaluación para que realice el test.
- 2.1.7. En la administración del sistema se puede ingresar, modificar o eliminar un test, motivos de consulta, fichas, información del test y agendar citas.
- 2.1.8. Para el ingreso del test se le muestra al usuario varios tríos de margaritas, estos tríos deben ser seleccionados y guardados con un nombre único de test. Para la modificación y eliminación del test se presentará al usuario un buscador en el cual se muestra todos los test ingresados previamente.

Al seleccionar un test, la información del mismo se despliega y se puede modificar o eliminar. Estos mismos pasos se siguen tanto para el test como para motivos de consulta.

- 2.1.9.** El ingreso de fichas será realizado de forma interactiva. Se le preguntará al sujeto evaluación sobre datos generales, datos actuales, se le hará una entrevista, etc. Para la modificación y eliminación de una ficha, se muestra al usuario un buscador donde están todas las fichas ingresadas previamente.
- 2.1.10.** Para la realización del test, el usuario selecciona el tipo de test que se va a desarrollar, busca al sujeto evaluación, se le muestran las instrucciones y comienza con el desarrollo del test.
- 2.1.11.** Los datos se guardan en el servidor de la Universidad del Azuay, para que de esta forma, estén siempre disponibles para los usuarios que tengan acceso al sistema, a cualquier hora, lugar y mediante cualquier dispositivo que tenga acceso a internet.
- 2.1.12.** El sistema tendrá su versión desarrollada para tablets Android e iOS, también se dispondrá de una página web para que computadoras con cualquier sistema operativo tengan acceso vía internet.
- 2.1.13.** En los dispositivos móviles los datos se almacenarán de manera local para evitar pérdida de información cuando la conexión a internet tenga intermitencias, posteriormente los datos se sincronizarán con la base de datos alojada en el servidor de la Universidad del Azuay, para que de esta manera cualquier otro dispositivo pueda consultar dicha información.

2.2. Requisitos No Funcionales

- 2.2.1. Usabilidad.-** El producto será fácil de usar para personas sin experiencia con las computadoras y tablets, además será capaz de adaptarse a las necesidades específicas de los usuarios y administradores. Para ayudar a la comprensión de usuarios y administradores, las aplicaciones mostrarán mensajes intuitivos, además la aplicación no podrá cerrarse hasta que sus operaciones sean concluidas.

- 2.2.2. Funcionalidad.-** El software será sometido a una etapa de pruebas, en la cual se detectarán posibles errores y se realizarán los cambios debidos, con la finalidad de que los usuarios y administradores puedan familiarizarse con software y queden complacidos.
- 2.2.3. Simplicidad.-** El producto será sencillo, de manera que no sea requerida la intervención de un especialista. La idea es desarrollar una herramienta sencilla, que no requiera mucho tiempo de adiestramiento para poderla usar.
- 2.2.4. Flexibilidad.-** Las aplicaciones deben ser capaces de crear, modificar o eliminar funciones, sin que se vea afectado su funcionamiento total.
- 2.2.5. Portabilidad.-** La página web deberá ejecutarse sin ningún problema en cualquier navegador web. Las aplicaciones móviles deberán ejecutarse de forma correcta tanto en Android como en iOS, y visualizarse sin errores en diferentes tamaños de pantalla de una Tablet o un Ipad.
- 2.2.6. Confiabilidad.-** El sistema tendrá la capacidad de evitar errores como resultado de haberse producido un fallo del sistema, además debe mantener un nivel de ejecución específico en caso de fallos del software o de infracción de sus interfaces específicas.

2.3. Referencias

R1. Función de Ingreso

Código	Requerimiento	Tipo
R1.1	Ingresar nombre de usuario y contraseña del Administrador.	Evidente
R1.2	Ingresar datos de identificación (datos generales y datos de padres) del Sujeto de Evaluación.	Evidente
R1.3	Ingresar motivo de la consulta, antecedentes personales, y datos del examen psiquiátrico.	Evidente

R1.4	Agendar una fecha para la cita con el Sujeto de Evaluación.	Evidente
R1.5	El almacenamiento de la información se hará de forma inmediata.	Oculto

Tabla 1: Requerimiento de función de ingreso

R2. Función Toma del Test

Código	Requerimiento	Tipo
R2.1	El administrador o el investigador, seleccionan el Test desarrollado por el Sujeto a Evaluar.	Evidente
R2.2	El administrador o el investigador, buscan y seleccionan el nombre del Sujeto de Evaluación para que pueda llenar el Test seleccionado previamente.	Evidente
R2.3	El test facilita la evaluación de los elementos de alerta, atención selectiva focalizada, atención sostenida y atención dividida.	Evidente
R2.4	Los resultados de la valoración de cada Sujeto Evaluación serán únicos y fáciles de identificar.	Oculto
R2.5	En la interfaz se identifica de forma clara cuál es el Sujeto Evaluación y qué Test está desarrollando. Durante el desarrollo del Test se tiene en cuenta la edad, experiencia tecnológica y las capacidades del Sujeto Evaluación para obtener un resultado general.	Evidente
R2.6	Medir variables de tiempo de respuesta total de la prueba, tiempo transcurrido entre la selección de cada ítem, tiempo tomado en responder a cada distractor.	Oculto
R2.7	Calcular de manera independiente el número de aciertos y errores, tanto del test como de los distractores.	Oculto

Tabla 2: Requerimientos de función toma del test

R3. Función de Resultados

Código	Requerimiento	Tipo
R3.1	Buscar y seleccionar al Sujeto de Evaluación.	Evidente
R3.2	Ingresar información sobre la consulta (motivo de la consulta, observaciones, tratamiento, etc.).	Evidente
R3.3	Mostrar un historial sobre los test realizados por los Sujetos Evaluados.	Evidente
R3.4	Mostrar los resultados obtenidos de cada test: aciertos, errores, omisiones, tiempo, etc.	Evidente

Tabla 3: Requerimientos de función de resultados

R4. Función de Manejo de Información

Código	Requerimiento	Tipo
R4.1	Mediante el módulo de configuración accesible por el Administrador se puede crear, modificar o eliminar las variaciones del Test de Margaritas.	Evidente
R4.2	Mediante el módulo de configuración accesible por el Administrador se puede crear, modificar o eliminar los motivos de consulta.	Evidente
R4.3	Registrar y almacenar los resultados del Test de Margaritas de cada Sujeto Evaluación.	Oculto

Tabla 4: Requerimientos de función de manejo de resultados

2.4. Diagrama de Actores

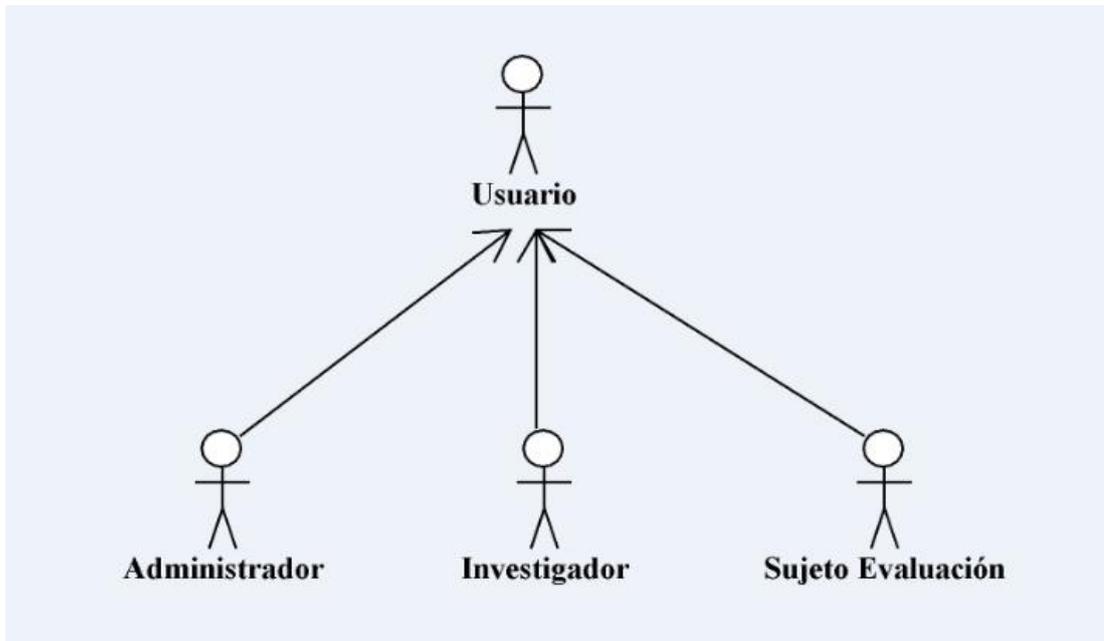


Ilustración 1: Diagrama de Actores

2.5. Diagrama de Casos de Uso

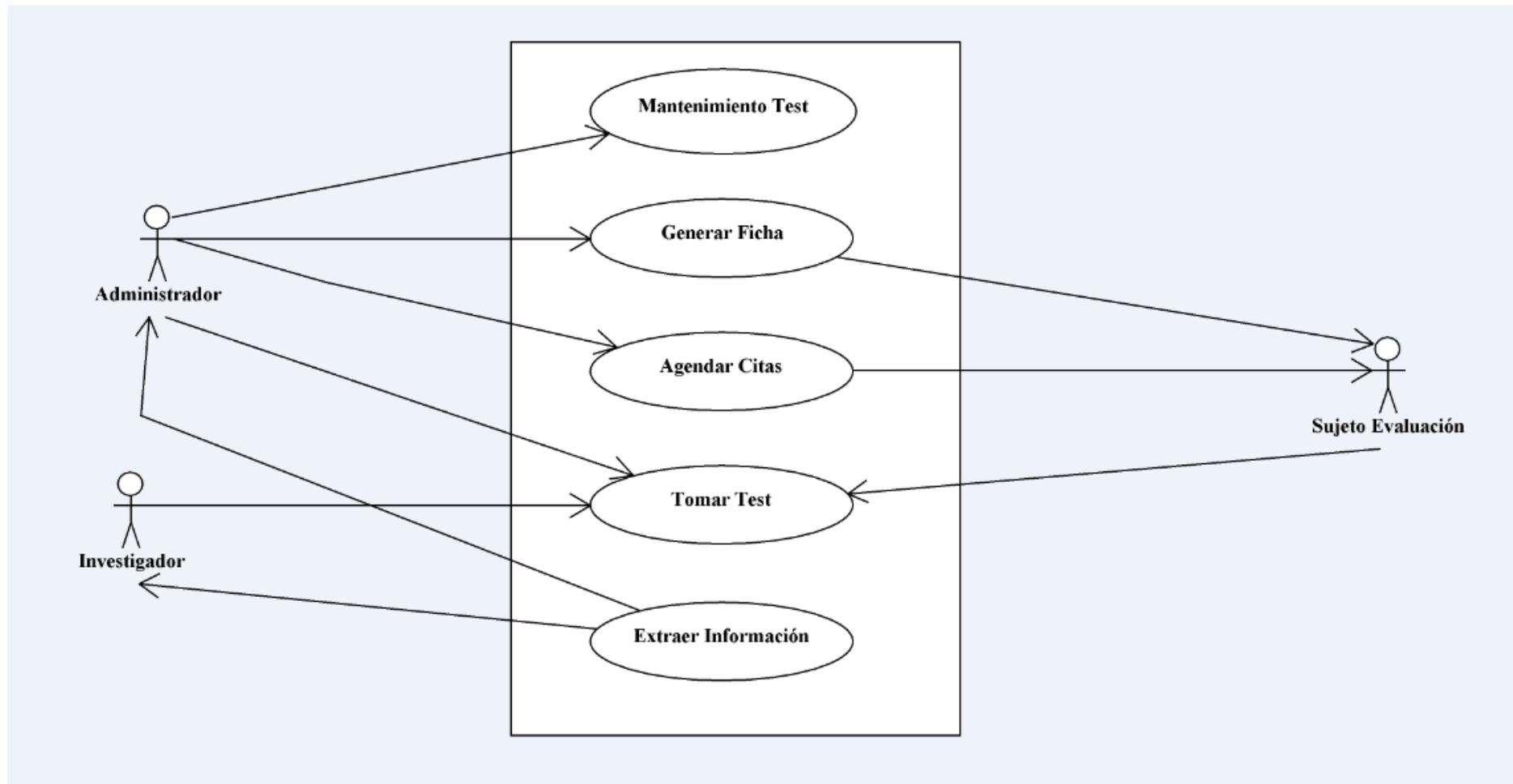


Ilustración 2: Diagrama de Casos de Uso

2.6. Especificación Casos de Uso

2.6.1. Actores

Actor	Administrador
Casos de Uso	Mantenimiento Test, Generar Ficha, Agendar Citas, Tomar Test, Extraer Información.
Descripción	Este actor debe estar registrado en el sistema para realizar cualquier gestión dentro del sitio.

Tabla 5: Actor Administrador

Actor	Investigador
Casos de Uso	Tomar Test, Extraer Información.
Descripción	Este actor necesita permiso del Administrador, además requiere de información ingresada previamente, para tomar el test y obtener resultados.

Tabla 6: Actor Investigador

Actor	Sujeto Evaluación
Casos de Uso	Tomar Test.
Descripción	Este actor, con permiso del Administrador o Investigador, puede realizar las gestiones en las que se encuentre involucrado.

Tabla 7: Actor Sujeto Evaluación

2.6.2. Casos de Uso

Caso Uso	Mantenimiento Test
Actores	Administrador
Propósito	Permitir crear, modificar y eliminar Test de Margaritas; además ingresar, modificar y eliminar Motivos de Consulta.

Resumen	Permite realizar mantenimiento del Test de Margaritas y de Motivos de Consulta.
Referencias Cruzadas	R1.1, R4.1, R4.2

Tabla 8: Mantenimiento Test

Caso Uso	Generar Ficha
Actores	Administrador, Sujeto Evaluación.
Propósito	Permitir crear, modificar o eliminar información del Sujeto Evaluación; así como ejecutar un control del seguimiento realizado ha dicho sujeto.
Resumen	Permite realizar mantenimiento de información y seguimiento del Sujeto Evaluación
Referencias Cruzadas	R1.2, R1.3

Tabla 9: Generar Ficha

Caso Uso	Agendar Citas
Actores	Administrador, Sujeto Evaluación.
Propósito	Permite ingresar una o varias citas en una fecha y horario establecidos, en la que se atenderá al Sujeto de Evaluación, además de mantener un historial de sujetos atendidos.
Resumen	Agendar citas con el Sujeto de Evaluación.
Referencias Cruzadas	R1.4

Tabla 10: Agendar Citas

Caso Uso	Tomar Test
Actores	Investigador, Administrador, Sujeto Evaluación.
Propósito	El Administrador o Investigador selecciona el nombre del Sujeto Evaluación y proporciona las indicaciones necesarias para que el Sujeto resuelva el test sin ningún problema.

Resumen	Indicaciones y desarrollo del Test de Margaritas.
Referencias Cruzadas	R2.1, R2.2, R2.5

Tabla 11: Tomar Test

Caso Uso	Extraer Información
Actores	Investigador, Administrador
Propósito	Tanto el Investigador como el Administrador, pueden obtener diferentes informes a partir de la información obtenida como resultado de los diferentes test aplicados.
Resumen	Extraer información de los test y obtener resultados.
Referencias Cruzadas	R3.1, R3.2, R3.3, R3.4

Tabla 12: Extraer Información

2.7. Diagrama de Clases

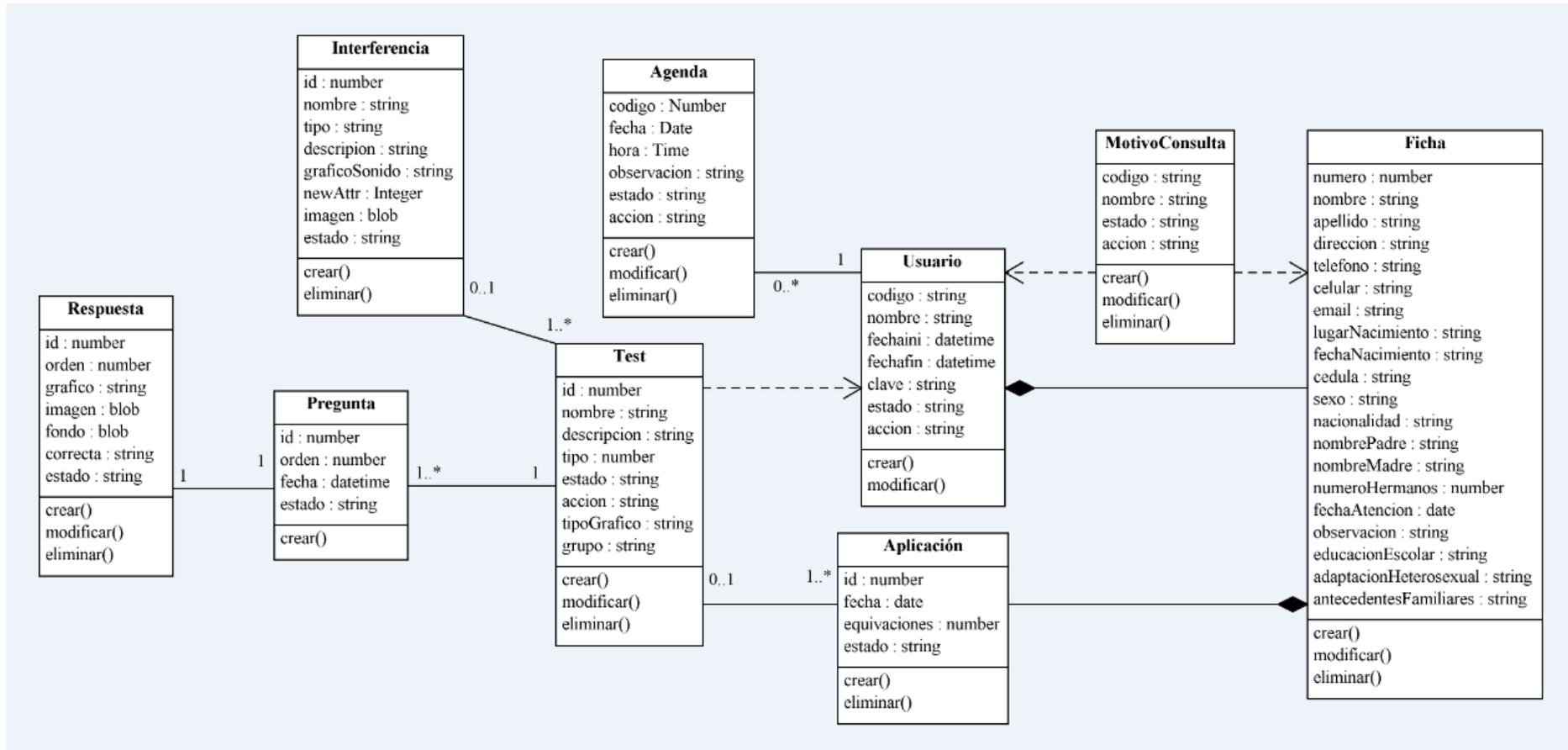


Ilustración 3: Diagrama de Clases

2.8. Diagrama de Secuencia

2.8.1. Diagrama de Secuencia – Mantenimiento Test

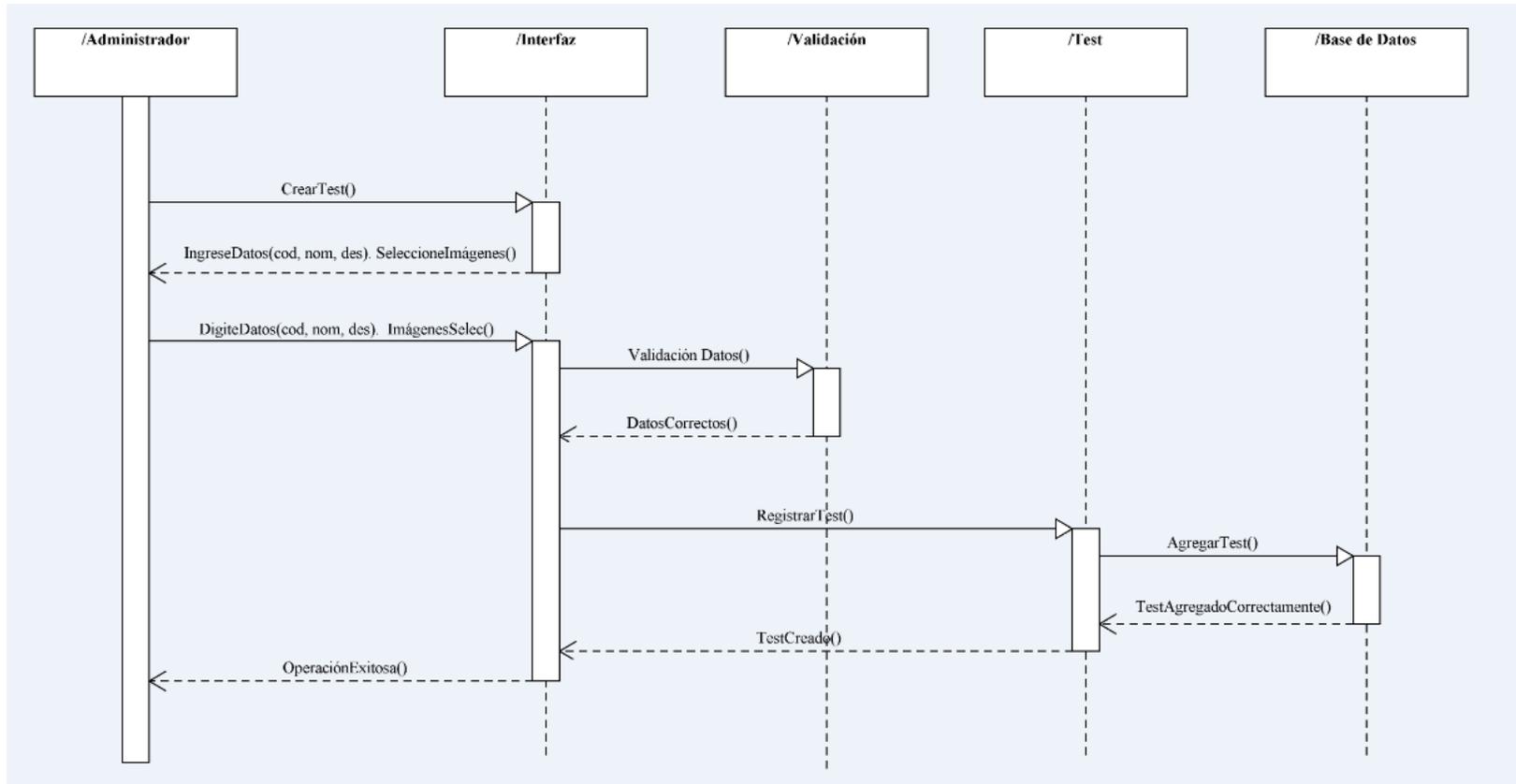


Ilustración 4: Diagrama de Secuencia – Mantenimiento Test

2.8.2. Diagrama de Secuencia – Generar Ficha

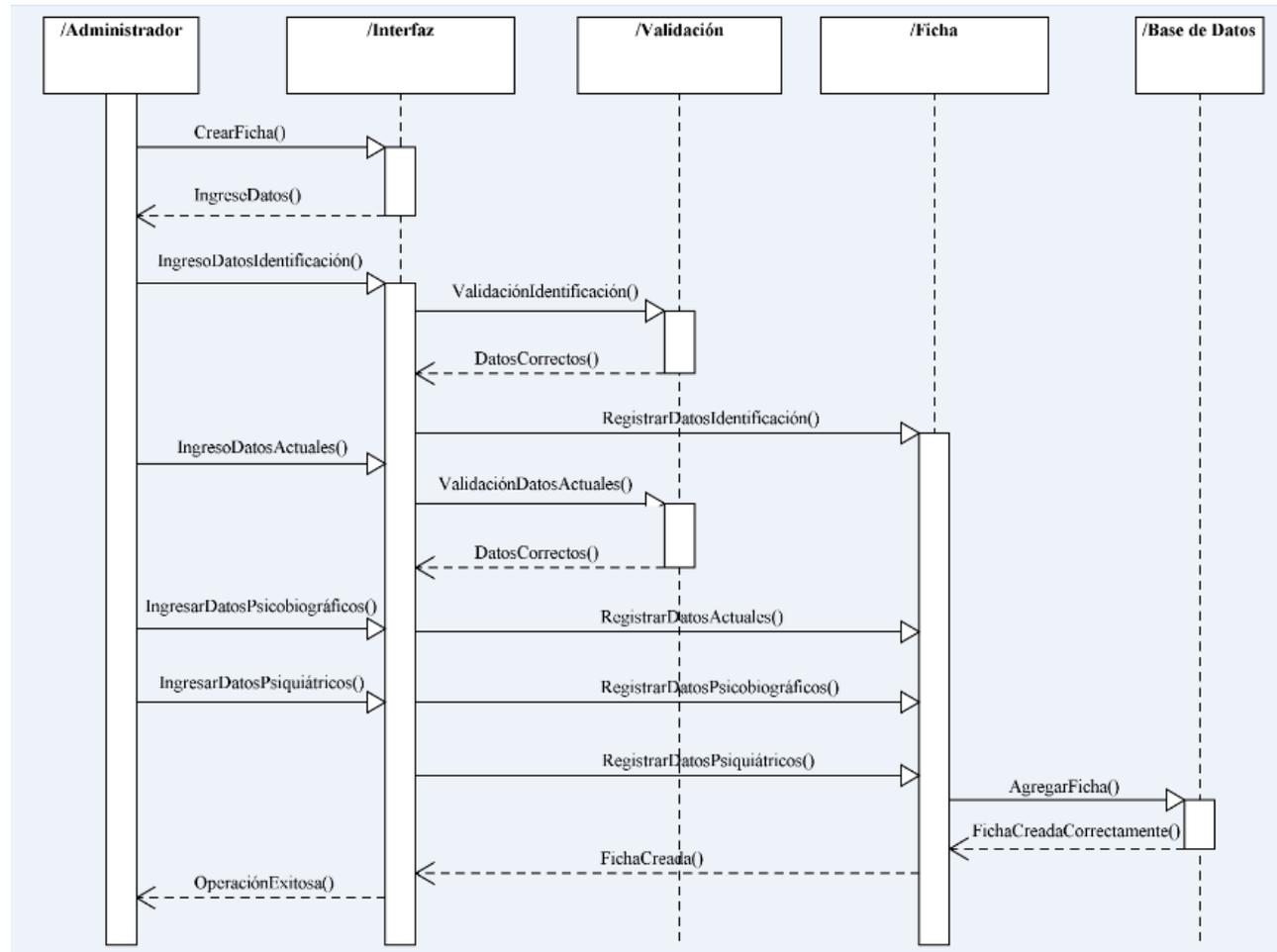


Ilustración 5: Diagrama de Secuencia – Generar Ficha

2.8.3. Diagrama de Secuencia – Agendar Citas

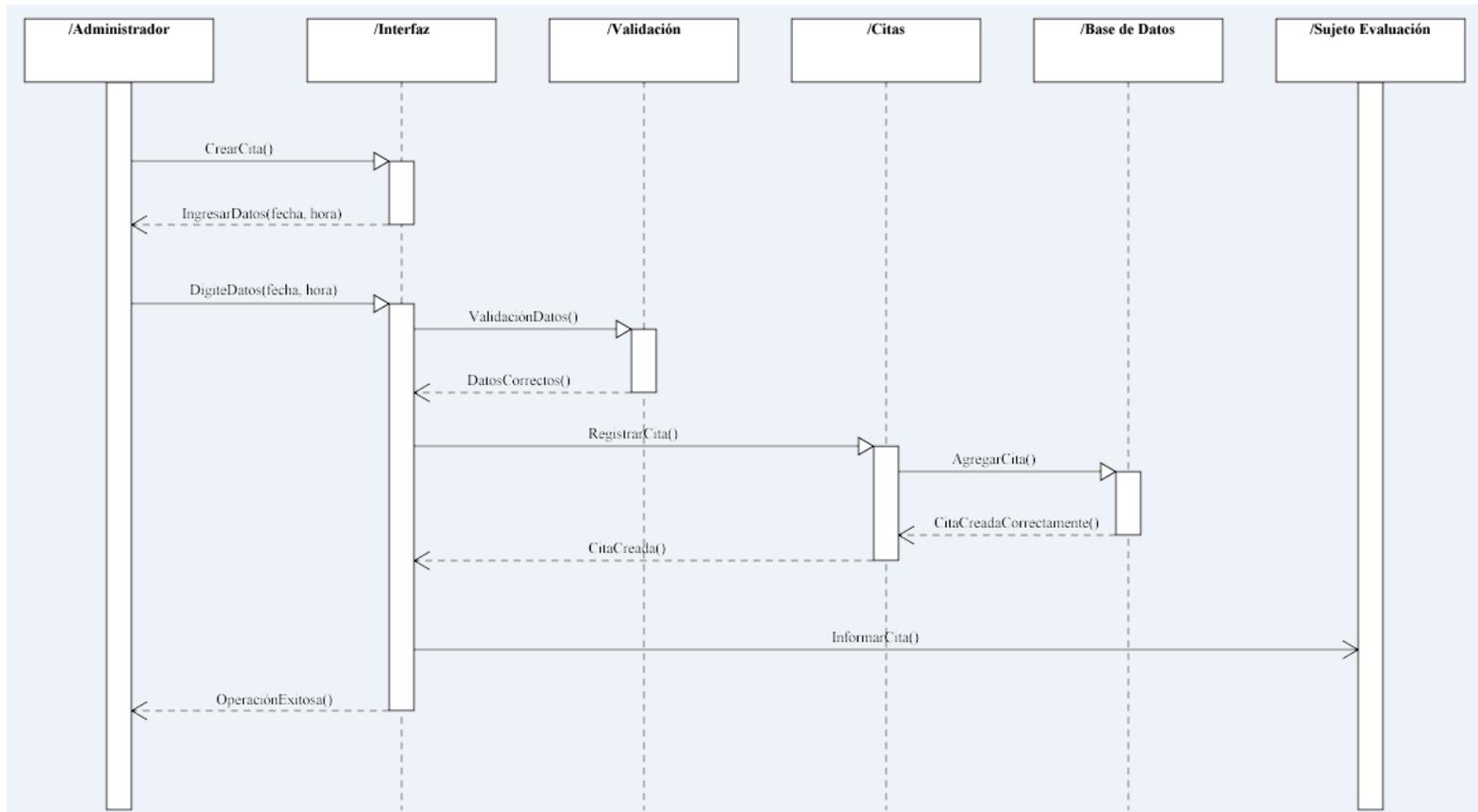


Ilustración 6: Diagrama de Secuencia – Agendar Citas

2.8.4. Diagrama de Secuencia – Tomar Test

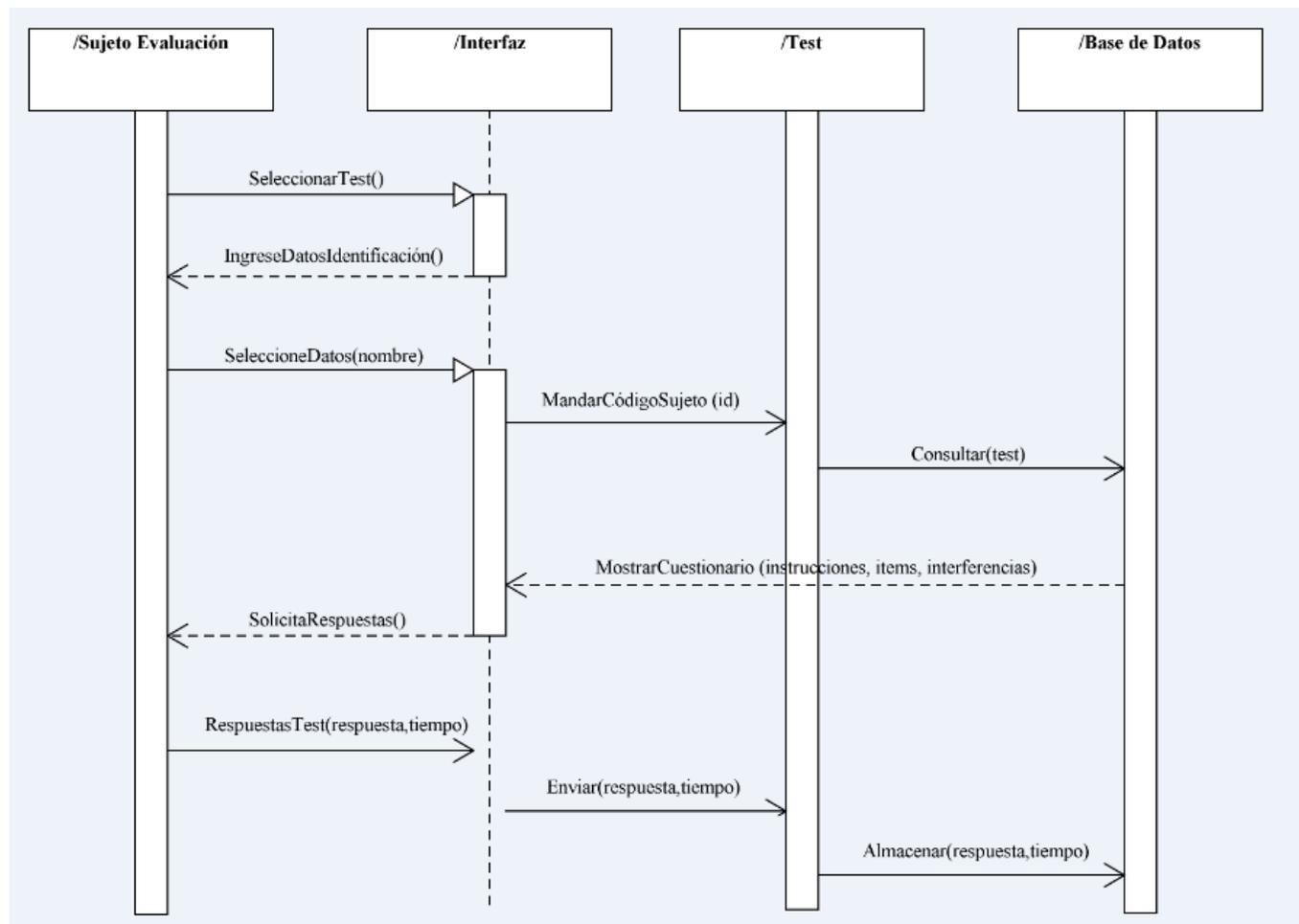


Ilustración 7: Diagrama de Secuencia – Tomar Test

2.8.5. Diagrama de Secuencia – Extraer Información

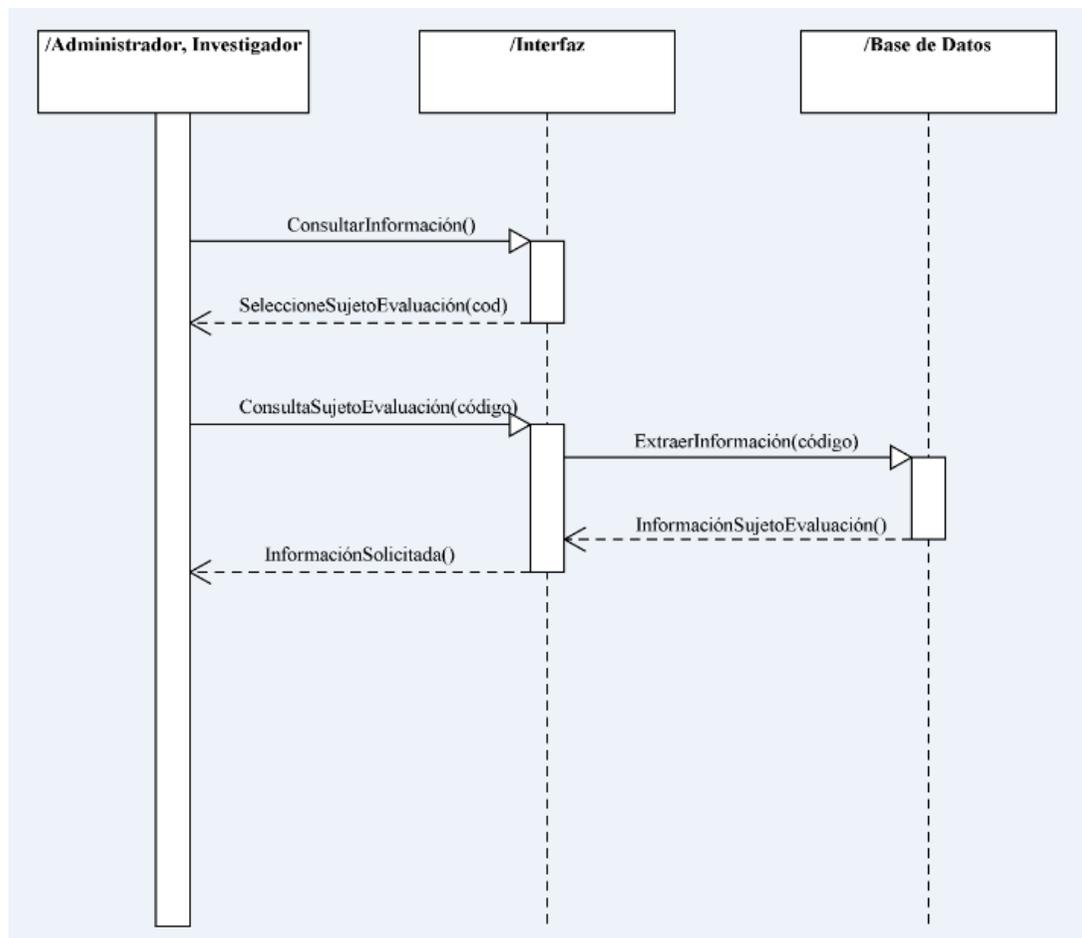


Ilustración 8: Diagrama de Secuencia – Extraer Información

CAPÍTULO 3: IMPLEMENTACIÓN

El gestor de Base de Datos utilizado es MySQL versión 5.1.66 en conjunto con SQLite versión 3.7.4 (aplicaciones móviles), además archivos PHP en formato JSON (para el intercambio de datos), para construir un Web Service que permita conectar las aplicaciones móviles con la base de datos principal (MySQL).

El desarrollo en la plataforma Android, se lo hace con el lenguaje de programación Java en conjunto con el SDK de Android, los cuales servirán para construir las clases y la interfaz de la aplicación, el IDE para el desarrollo del producto es Eclipse versión Kepler Release.

En la plataforma iOS, para construir las clases e interfaces se utiliza el lenguaje de programación Objective C, en conjunto con iOS SDK y el IDE de desarrollo es XCode versión 5.

Para el diseño gráfico de la página web, se utiliza el lenguaje de programación HTML5 en conjunto con CSS3, y para la parte funcional el lenguaje PHP en conjunto con javascript.

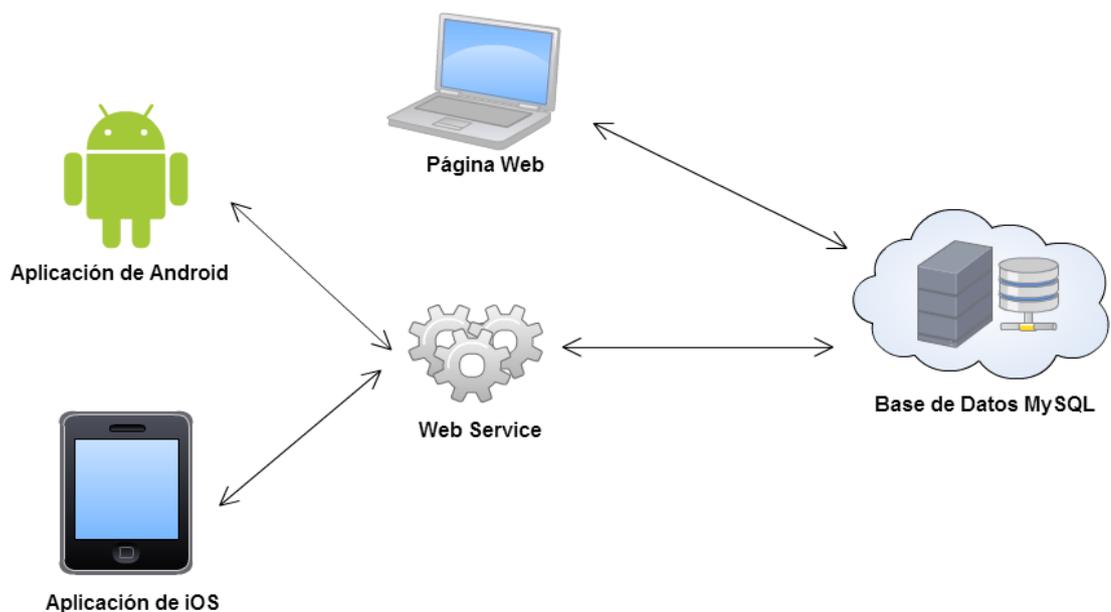


Ilustración 9: Arquitectura Test de Margaritas

3.1. Desarrollo Base de Datos

Se administra dos Bases de Datos, una en MySQL (principal) y otra en SQLite (secundaria).

3.1.1. Definición de MySQL

MySQL es un sistema de administración de base de datos relacionales (SGBDR) rápido, robusto y fácil de usar. Es ideal para administrar datos en un entorno de red (arquitectura cliente/servidor) y permite compatibilidad con el servidor de páginas Web Apache y el lenguaje de páginas web PHP. Es fácil de personalizar y adaptar, puesto que es Open Source. (Cyril THIBAUD, 2006)

3.1.2. Explicación de MySQL

El motor de almacenamiento que se utiliza en MySQL es MyISAM. La característica principal de MyISAM es la gran velocidad que se obtiene al realizar consultas, puesto que no se hacen comprobaciones de la integridad referencial, ni tampoco se bloquean las tablas para realizar las operaciones ya que hay ausencia de características de atomicidad. (Hinz, y otros, MySQL Documentation, 2014)

La información almacenada en MySQL es accedida a través de la página web y por los dispositivos móviles con la utilización de Web Services. En esta base de datos no se permite la eliminación física de ningún campo, se hace una eliminación lógica utilizando el campo llamado ESTADO (que se encuentra en cada tabla), que cambia de “Normal” a “Anulado” para que la información no sea visible en la página web o en los dispositivos móviles.

La primera vez que se instala la aplicación móvil en los dispositivos, el aplicativo de forma automática descarga todas las tablas necesarias de MySQL y duplica esta información a la base de datos SQLite propia de cada dispositivo. Este paso es necesario para que la aplicación pueda manejarse de forma independiente y funcione sin tener una conexión a internet.

Una vez que se tenga descargada la información a SQLite, se debe tener en consideración que, el usuario mediante la página web puede modificar información de las tablas que son necesarias para el funcionamiento de la aplicación (por ejemplo: TPA_TEST, TPA_PREGUNTA, TPA_RESPUESTA, etc), estas tablas deben ser actualizadas en SQLite.

Para actualizar los datos se deben comparar las tablas de MySQL con las de SQLite y verificar tabla por tabla si se ha modificado su contenido. Este proceso toma demasiado tiempo y hace que la aplicación se vuelva lenta, para solucionar este problema se optó por manejar TRIGGERS en ciertas tablas. Cuando se ingresa o modifica alguna tabla en MySQL se dispara un TRIGGER, este TRIGGER crea un nuevo campo en la tabla REG_MOVIMIENTOS, en el campo se almacena: el nombre de la tabla donde se realizó el cambio, la acción que se realizó (INSERT, UPDATE) y la sentencia SQL. Cuando la base de datos SQLite quiera actualizar su información, lo único que hace es comparar la tabla REG_MOVIMIENTOS de MySQL con la suya, si existen cambios los ejecuta uno por uno.

Para realizar el proceso de actualización, el dispositivo móvil tiene un botón llamado ACTUALIZAR, cuando el usuario presiona este botón se realiza todo el proceso antes mencionado de forma transparente.

3.1.3. Definición de SQLite

SQLite es un sistema de administración de base de datos relacionales (SGBDR). Es de código abierto, sus principales características son: no necesita de ningún servidor ya que escribe y lee archivos de discos normales directamente, no necesita de mucha configuración, ocupa muy poco espacio de almacenamiento y es multiplataforma.

3.1.4. Explicación de SQLite

Para los dispositivos móviles (Android, iOS) se utiliza la Base de Datos SQLite, que es propia del dispositivo, se trabaja con esta base de datos local para no depender de una conexión a internet durante el uso de la aplicación.

La Base de Datos SQLite descarga la información que necesita y además genera información que luego comparte con MySQL, para estos procesos, es necesario contar con conexión a internet.

Una vez que la información pase de SQLite a MySQL ciertas tablas de SQLite se vaciarán para evitar que esta base de datos se llene de información que ya no será útil, el usuario decide en que momento va a eliminar estos datos.

3.2. Desarrollo Android

Esta aplicación está desarrollada para tablets de 7 pulgadas, que cuenten con un sistema operativo Android de versión superior a 3.0 – API Level 11 (Honeycomb).

3.2.1. Definición del Sistema Operativo Android

Según (Robledo Sacristan & Robledo Fernández, 2012): *“Android es un sistema operativo basado en Linux, que es un núcleo de sistema operativo libre, gratuito y multiplataforma. Este sistema operativo permite programar aplicaciones empleando una variación de Java llamada Dalvik, y proporciona todas las interfaces necesarias para desarrollar fácilmente aplicaciones que acceden a las funciones del teléfono.”*

3.2.2. Arquitectura del Sistema Operativo Android

Existen cinco componentes principales que conforman el sistema operativo Android, estos son: Aplicaciones, Framework de aplicaciones, Android Runtime, Librerías, Kernel de Linux.

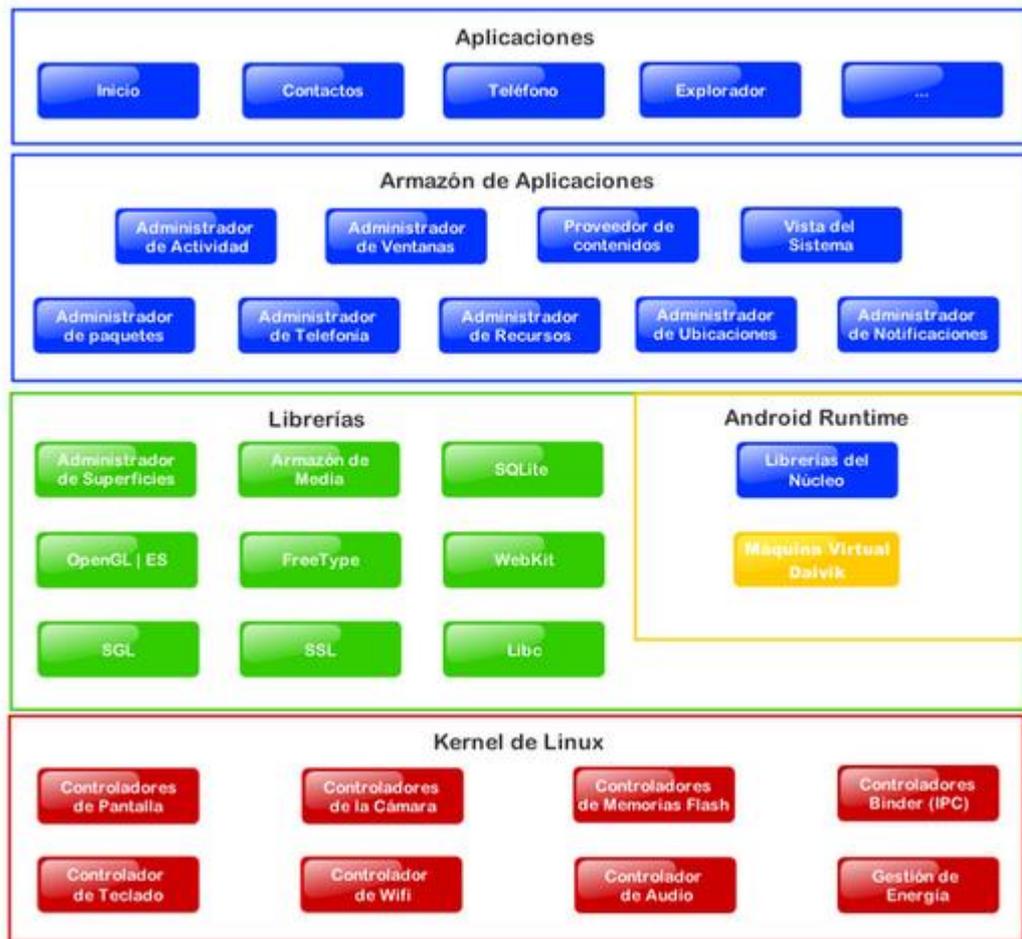


Ilustración 10: Capas Android

Fuente de la Imagen: (Cancela García & Ostos Lobo, 2014)

- **Aplicaciones.**- Esta capa incluye todas las aplicaciones del dispositivo, es decir, las que vienen preinstaladas en el dispositivo y las que el usuario ha ido añadiendo posteriormente. Todas las aplicaciones utilizan los siguientes servicios y sistemas: Views (conjunto de componentes), Content Providers (proveedor de contenidos), Resource Manager (gestor de recursos), Notification Manager (gestor de notificaciones) y Activity Manager (gestor de actividades). (Cancela García & Ostos Lobo, 2014)
- **Armazón de aplicaciones (Framework).**- Esta capa contiene todas las clases y servicios que se utilizan para que las aplicaciones se ejecuten correctamente; su finalidad es reutilizar componentes, es decir, cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede utilizar estas capacidades. (Cancela García & Ostos Lobo, 2014)

- **Librerías.-** Esta capa corresponde al conjunto de librerías que utiliza Android. Estas librerías están escritas en C o C++, el objetivo principal es proporcionar funcionalidad a las aplicaciones, para que las tareas que se repiten con frecuencia puedan ser reutilizadas, garantizando que se lleven a cabo de forma eficiente. (Cancela García & Ostos Lobo, 2014)
- **Android Runtime.-** Este entorno no es considerado como un nivel, ya que tiene librerías. El componente principal es una máquina virtual de Java llamada Dalvik, esta permite correr múltiples máquinas virtuales de forma eficiente, Android corre su propio proceso con su propia instancia de máquina virtual. (Cancela García & Ostos Lobo, 2014)
- **Kernel de Linux.-** Android utiliza el núcleo de Linux 2.6 como capa de abstracción entre el hardware y el resto de las capas de la arquitectura, puesto que contiene los drivers necesarios para ejecutar cualquier componente de hardware. El kernel además se encarga de gestionar la energía, memoria, seguridad, etc. del teléfono. (Cancela García & Ostos Lobo, 2014)

3.2.3. Estructura de un Proyecto Android

Los proyectos de Android son estructurados en un fichero .apk que se instala en el dispositivo móvil. Contienen directorios y archivos, algunos se crean de forma automática mientras que otros los vamos creando según las necesidades. Los directorios y archivos que ocupamos en el Test de Margaritas son:

- **src/:** Esta carpeta contiene todo el código fuente de la aplicación. La carpeta principal del Test de Margaritas es src/app/testmargarita.
- **bin/:** Directorio de salida de la compilación. Aquí es donde se puede encontrar el archivo .apk final y otros recursos compilados. (Android Developers, 2014)
- **gen/:** Esta carpeta contiene el código generado de forma automática por el SDK. Los ficheros que contiene esta carpeta nunca deben ser modificados de forma manual. (Gironés, 2013)

- **assets/:** Esta carpeta está vacía. Se puede utilizar para almacenar archivos de activos brutos. Los archivos que se guardan aquí se compilan en un archivo .apk como está, y el nombre de archivo original se preserva. (Android Developers, 2014)
- **res/:** Esta carpeta contiene recursos de la aplicación como:
 - **drawable/:** Almacena imágenes (.png, .jpg) y archivos .xml que describen formas Drawable u objetos Drawable que contienen varios estados (normal, presionado, o el foco). Se dividen en drawable-hdpi, drawable-mdpi, drawable-xhdpi, para ubicar las imágenes dependiendo de la resolución del dispositivo.
 - **layout/:** Contiene ficheros .xml que permiten configurar las diferentes pantallas de la interfaz gráfica. En la aplicación del test las carpetas se dividen en layout-sw600dp/ y layout-sw600dp-land, con la utilización de sw600dp se consigue que la aplicación funcione para tablets de 7” en adelante, es decir, la aplicación solo se va a instalar si el ancho mínimo es igual o superior a 600dp. Con sw600dp logramos que el diseño sea multipanel (horizontal o vertical); mientras que con sw600dp-land el diseño es de panel único (horizontal). (Android Developers, 2014)
 - **menu/:** Contiene ficheros .xml que definen el menú de la aplicación.
 - **values/:** Permite cambiar los valores sin tener que abrir el código fuente, contiene los siguientes recursos:
 - **color/:** Se les asignan nombres a los colores en código hexadecimal.
 - **dimens/:** Tamaños de los textos, tablas, gráficos, etc.
 - **strings/:** Cadenas de texto.
 - **styles/:** El estilo que va a utilizar la aplicación.
- **libs/:** En esta carpeta se agregan nuevas librerías, según la necesidad de la aplicación.
- **AndroidManifest.xml:** Describe los aspectos principales de la aplicación y cada uno de sus componentes. Aquí se especifica el nombre, el icono, la versión de la aplicación, la versión mínima del SDK y los permisos que se van a ocupar, estos son:

- android.permission.INTERNET: Permite que la aplicación pueda subir y descargar información de la red.
- android.permission.WRITE_EXTERNAL_STORAGE: Permite leer y escribir en el microSD externo del dispositivo.
- android.permission.ACCESS_NETWORK_STATE: Informa sobre el estado de la conexión a internet.

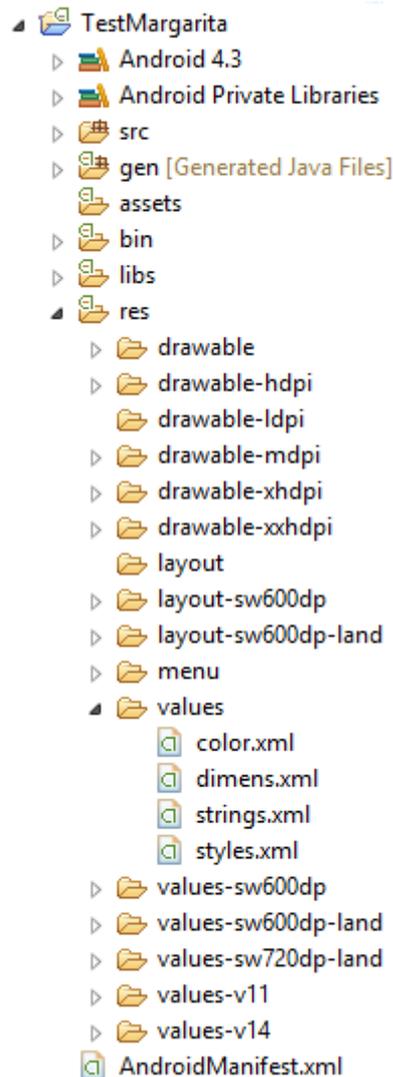


Ilustración 11: Estructura del Test de Margaritas, visualizado desde Eclipse

3.2.4. Explicación de Android

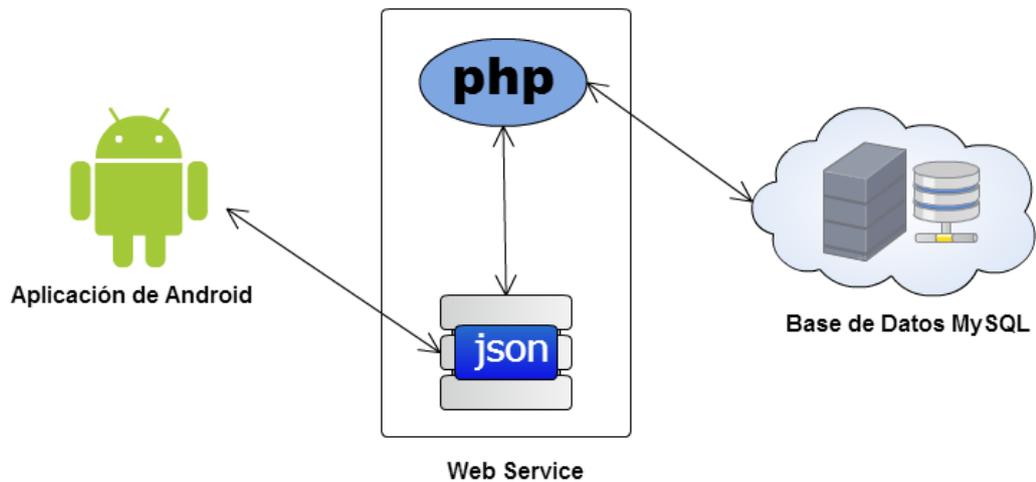


Ilustración 12: Arquitectura Test de Margaritas - Android

La aplicación fue desarrollada en el lenguaje de programación Java con el SDK de Android, puesto que es software libre y los dispositivos son más accesibles en el entorno, con precios cómodos.

El objetivo de la aplicación es que los niños (sujeto evaluación) puedan realizar el test con mayor facilidad, ya que para ellos es más cómodo adaptarse a una pantalla táctil, en comparación con un ratón de computadora.

Con esta aplicación el usuario (investigador) puede acceder a fichas, crear una nueva ficha, editar fichas, tomar el test, descargar fichas por escuela y sincronizar la información básica para el funcionamiento correcto; la aplicación presente en las tablets, se utiliza para recolectar información y no para analizar la información (esta tarea se realiza en la Página Web).

Por motivos de seguridad a cada usuario se le asigna previamente (en la Página Web) un usuario y una contraseña, en el momento que se inicia la aplicación se piden estos datos y se verifican a través de un Web Service si existe o no el usuario. Si el usuario existe se procede a crear una ficha con los datos del Sujeto a Evaluar, una vez

terminado este paso el usuario procede a Tomar el Test y por último ocupa el botón Sincronizar – Subir Información para actualizar los datos en MySQL.

La primera vez que se accede a la aplicación en Android, es necesario que el dispositivo cuente con conexión a internet, ya que se necesita descargar información de la base de datos principal para su funcionamiento. Después de descargar esta información, se puede acceder a la aplicación sin importar si dispone o no, de conexión a internet.

La aplicación esta subida al Play Store con el nombre Test de Margaritas.

3.3. Desarrollo iOS

3.3.1. Definición del Sistema Operativo iOS

El sistema operativo iOS, antes conocido como iPhone OS, es un sistema basado en Unix y fue creado por Apple Inc. Inicialmente fue desarrollado para dispositivos móviles iPhone, actualmente otros dispositivos de la misma empresa ocupan este sistema operativo: iPad, iPod Touch y Apple TV. La interfaz de iOS se basa en gestos multitáctiles que hace que la interacción con el dispositivo sea más natural y creativa. (Fernández Pérez, 2013)

3.3.2. Arquitectura del Sistema Operativo iOS

Existen cuatro componentes principales que conforma el sistema operativo de iOS, estos son: Core OS, Core Services, Media, Cocoa Touch.

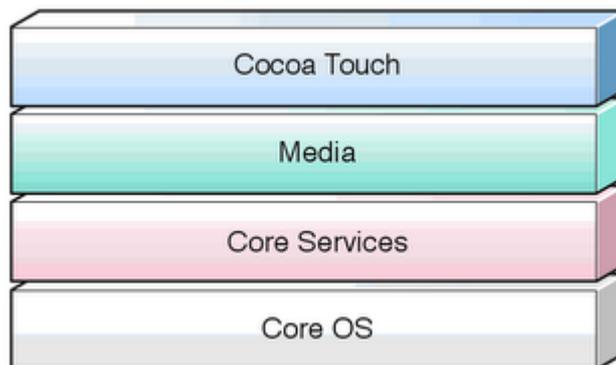


Ilustración 13: Capas iOS

Fuente de la imagen: (iOS Developer, 2014)

- **Cocoa Touch.-** Esta capa contiene frameworks claves para el desarrollo de aplicaciones iOS. También proporciona la infraestructura de aplicación básica y brinda soporte para tecnologías primordiales como la multitarea, entrada de datos táctil, notificaciones y muchos servicios de alto nivel del sistema. (iOS Developer, 2014)
- **Media (Medios de Comunicación).-** Esta capa permite poner en práctica la experiencia multimedia en las aplicaciones, es decir, contiene los gráficos, audio, video y tecnologías orientadas a mejorar la misma. Las tecnologías que posee esta capa, hace que sea más fácil la construcción de aplicaciones que se vean bien y suenen bien. (iOS Developer, 2014)
- **Core Services (Núcleo del Servicio).-** Esta capa contiene los servicios fundamentales del sistema que usan todas las aplicaciones. Esta capa también contiene tecnologías individuales como la ubicación, iCloud, medios sociales, SQLite, conexión a internet, etc. (iOS Developer, 2014)
- **Core OS (Núcleo del Sistema Operativo).-** Esta capa incluye características de bajo nivel, estas características son accedidas comúnmente por los frameworks de capas superiores. Se encarga de la gestión de memoria, el sistema de ficheros, la conexión a la red, la seguridad y los procesos de comunicación con el hardware externo. (iOS Developer, 2014)

3.3.3. Explicación de iOS

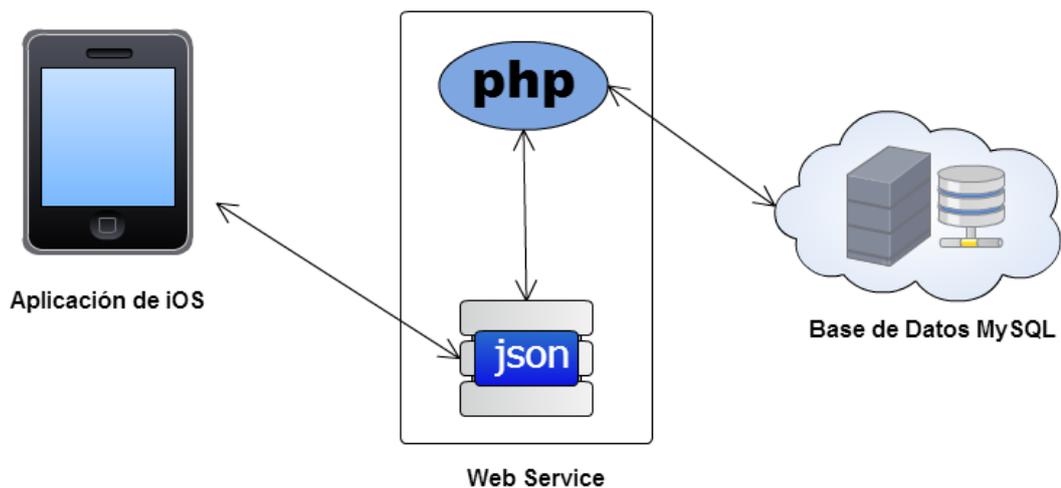


Ilustración 12: Arquitectura Test de Margaritas - iOS

Las aplicaciones iOS se pueden desarrollar únicamente en el IDE “XCode”, el cual está disponible solo para el Sistema Operativo MacOSX.

Esta aplicación está destinada a correr en tablets iPad de 10 pulgadas, pero por el momento no se la publicará en la tienda virtual de Apple, se mantendrá una versión Beta de la misma.

Con esta aplicación el usuario (investigador) puede ver las fichas y test ingresados desde la web, también puede aplicar tests con o sin interferencias y tests de una o tres filas de margaritas.

Esta aplicación a diferencia de la de Android, consulta y guarda la información directamente a la base de datos alojada en el servidor de la universidad a través de un webservice, esta no guarda datos localmente, razón por lo cual deberá estar conectada a internet en todo momento.

3.4. Desarrollo Web

3.4.1. Conceptos Utilizados

- **HTML.-** Acrónimo de HyperText Markup Language (lenguaje de marcado de hipertexto). Es un lenguaje de etiquetas que sirve para la elaboración de páginas web, define una estructura como texto, imágenes, videos, etc. Funciona como una de las piedras angulares de la World Wide Web. (Damián, 2011)
- **CSS3.-** Acrónimo de Cascading Style Sheets (hojas de estilo en cascada). Permite definir estilos visuales a los elementos del documento, como color, tamaños, tipo de letra, fondo, border, etc. Con las hojas de estilo se evitan los errores causados por la repetición, puesto que se define un estilo una sola vez y se lo puede reutilizar las veces que se desee. (AUBRY, 2012)
- **PHP.-** Acrónimo de Hypertext Preprocessor (procesador de hipertexto). Es un lenguaje de código abierto interpretado, de alto nivel, especialmente adecuado para el desarrollo web; puede ser incrustado en páginas HTML y ejecutado en

un servidor. Este lenguaje se caracteriza porque es solo interpretado pero no compilado. (PHP Documentation Group, 2014)

- **JavaScript.-** Es un lenguaje de programación interpretado orientado a objetos, ha sido desarrollado para usarse junto a otras herramientas Web. Está diseñado para trabajar en conjunto con HTML, para permitir mejoras en la interfaz de usuario y páginas web dinámicas. (Brooks, 2011)

3.4.2. Explicación de la parte Web

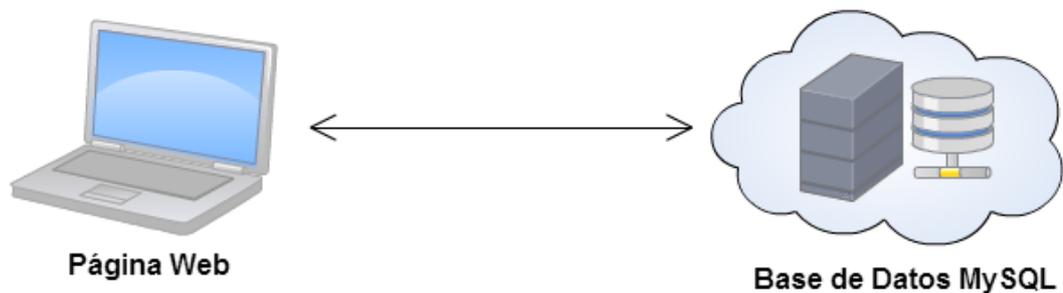


Ilustración 13: Arquitectura Test de Margaritas - Página Web

La aplicación web está desarrollada en los lenguajes: PHP, javascript y HTML sin el soporte de un IDE, es decir con código puro. Se puede acceder a la web del test de margaritas desde cualquier lugar que disponga de conexión a internet a través de cualquier explorador a excepción de Internet Explorer.

La web es la única de las tres plataformas del Test de margaritas (Web, iOS y Android) que no usa un middleware para realizar consultas a la Base de Datos, si no que se comunica directamente con esta.

La aplicación está dividida en componentes (capas), los cuáles cumplen diferentes funciones.

Según (Niekamp): *“Un componente de software es un elemento de un sistema que ofrece un servicio predefinido, y es capaz de comunicarse con otros componentes.”*.

Para el test de margaritas, los componentes usados son los siguientes:

- **Datos:** Recibe una petición por parte del usuario. Se encarga de preparar la información antes de realizar la acción o proceso indicado por el usuario. Este componente recibe, valida, los datos y los pasa a Persistencia. Además comprueba si el usuario que intenta realizar un proceso tiene permiso para hacerlo.
- **Persistencia:** Toma la información que envió el componente Datos, la procesa dependiendo de lo que requiera el usuario y envía el resultado al siguiente componente. Este es el único componente que se comunica con la base de datos.
- **Vista:** Toma la información resultante de un proceso y la muestra al usuario de manera que este la pueda interpretar.

En el archivo “Componentes.php” están las funciones necesarias para ejecutar dichos componentes. A continuación se indica la función usada para ejecutar los componentes en la web:

```
function ejecutar_componentes($evento,$cont){
    $modulos = array("Datos","Persistencia","Vista");
    foreach ($modulos as $modulo):
        require_once $modulo.".php";
        $obj = new $modulo;
        $cont = $obj->$evento($cont);
        if (!$cont)
            die("Ha ocurrido un error! No se puede continuar");
    endforeach;
}
```

Por ejemplo, cuando el usuario da clic en el botón iniciar sesión, se envía una petición al componente Datos, este recibe el nombre de usuario y contraseña, encripta la contraseña y envía dicha información a Persistencia, este se encarga de comprobar contra la Base de Datos si la combinación de usuario y contraseña son o no correctos y envía el resultado a Vista, en caso de ser positivo el resultado, Vista envía un mensaje de bienvenida, caso contrario envía un mensaje de error.

A continuación una fracción del código del archivo “test.php”, en el cual se envía el evento “iniciar_sesion”:

```
require_once 'Componentes.php';
require_once 'Container.php';
$cont = new Usuario;
$evento = 'iniciar_sesion';
ejecutar_componentes($evento,$cont);
```

La variable \$cont es una instancia de la clase Usuario, esta clase está declarada en el archivo “Container.php”, se envía esta variable junto con el evento correspondiente, como parámetros de la función ejecutar_componentes, dicha función se encarga de ejecutar este evento en los componentes: Datos, Persistencia y Vista.

Se eligió este paradigma de programación ya que es el que más se ajusta a nuestros intereses, ya que al tener la posibilidad de adherir o quitar componentes, tenemos la opción de usar el mismo código para diferentes causas. En este caso, la causa principal fue reutilizar código de la web en el Web Service para las apps móviles.

Como se mencionó anteriormente, el Web Service que usan las apps móviles del Test de Margaritas está desarrollado en PHP, por lo que se pudo reutilizar el código escrito para la página web, usando únicamente dos de los tres componentes indicados anteriormente: Datos y Persistencia. Al ser este un middleware, no se necesitan mostrar los datos, por lo cual no se usa el componente Vista, el dispositivo móvil que haya realizado la petición es el encargado de mostrar la información al usuario.

A continuación se indica la función usada para ejecutar los componentes utilizados por el Web Service.

```
function ejecutar_componentes_webservice($evento,$cont){
    $modulos = array("Datos","Persistencia");
    foreach ($modulos as $modulo):
        require_once $modulo.".php";
        $obj = new $modulo;
        $cont = $obj->$evento($cont);
    if (!$cont)
```

```
die("Ha ocurrido un error! No se puede continuar");  
endforeach;  
}
```

3.5. Manuales de Usuario – Web

URL: http://lab.uazuay.edu.ec/test_atencion/

Usuario: prueba

Contraseña: prueba

La página web del Test de Margaritas, puede ser accedida desde un navegador de internet. La dirección URL es http://lab.uazuay.edu.ec/test_atencion/.

El usuario debe ingresar a su navegador web (Google Chrome, Mozilla Firefox, Safari) y escribir la dirección URL de la página web. Una vez cargada la página se visualiza una pantalla donde se le solicita la usuario ingresar datos de autenticación (usuario, contraseña y captcha).



Ilustración 14: Página Web – inicio de sesión

3.5.1. Iniciar Sesión

- Para comenzar a utilizar el Test de Margaritas, el usuario debe ingresar nombre de usuario, contraseña y dibujar la forma presentada; si esta forma es muy difícil se presiona la opción nueva forma y se genera una nueva imagen. Si se visualiza el mensaje: “Captcha passed”, presione el botón Iniciar Sesión.



Ilustración 15: Página Web – captcha correcto

- Si los datos ingresados por el usuario no son correctos se visualiza el siguiente mensaje: “Nombre de usuario o contraseña incorrectos”.

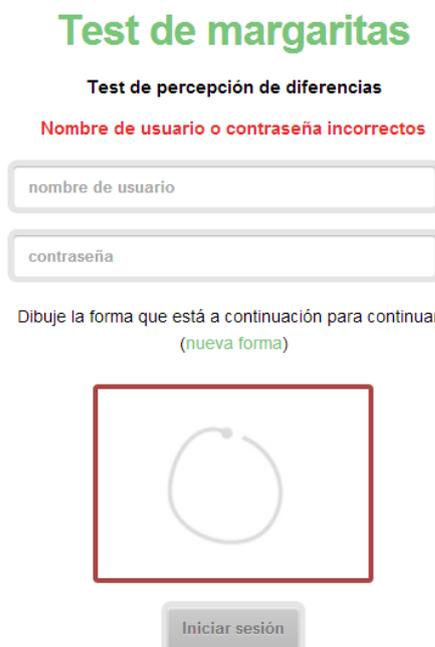


Ilustración 16: Página Web – mensaje: Nombre de usuario o contraseña incorrectos

- Si el nombre de usuario, contraseña y captcha son correctos, el sistema le presenta al usuario un menú con las siguientes opciones: Home, Administración, Procesos, Tomar Test, Acerca de y Usuario. Por motivos de seguridad la opción Administración únicamente se presenta en usuarios administradores, esta opción no es visible para los demás tipos de usuario.



Ilustración 17: Página Web – pantalla de bienvenida, menú principal

3.5.2. Menú Usuario

- Cuando el usuario presiona el menú Usuario, se presentan las siguientes opciones: Mi Perfil, Cambiar Contraseña, Salir

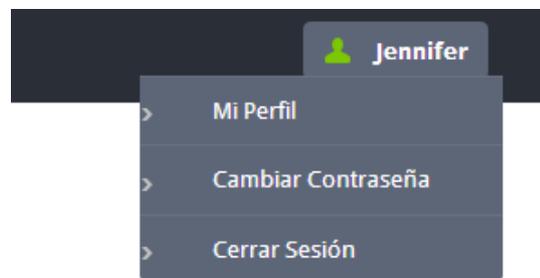


Ilustración 18: Página Web – submenú usuario

3.5.2.1. Mi Perfil

- En la opción Mi Perfil, el usuario puede modificar los datos que crea necesarios. Una vez haya terminado, almacena la información de perfil presionando el botón Guardar.

Mi perfil

Puede modificar sus datos de ser necesario

Puede modificar sus datos de ser necesario

(*) CAMPOS OBLIGATORIOS

CÉDULA: 111111111

PRIMER NOMBRE:*

SEGUNDO NOMBRE:

APELLIDO PATERNO:*

APELLIDO MATERNO:

E-MAIL:

NOMBRE DE USUARIO:*

DURACIÓN DE CADA CITA (MINUTOS)

TIPO DE USUARIO: ADMINISTRADOR

Ilustración 19: Página Web – modificación de los datos de perfil

- Si la información almacenada es correcta se visualiza el siguiente mensaje: “El usuario se ha modificado correctamente”.

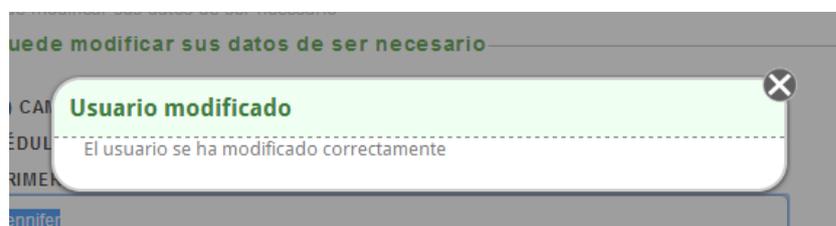


Ilustración 20: Página Web – mensaje: El usuario se ha modificado correctamente

3.5.2.2. Cambiar Contraseña

- En la opción Cambiar Contraseña, el usuario puede modificar su contraseña digitando la contraseña actual e ingresando una nueva contraseña. Una vez haya terminado, almacena la información presionando el botón Guardar.

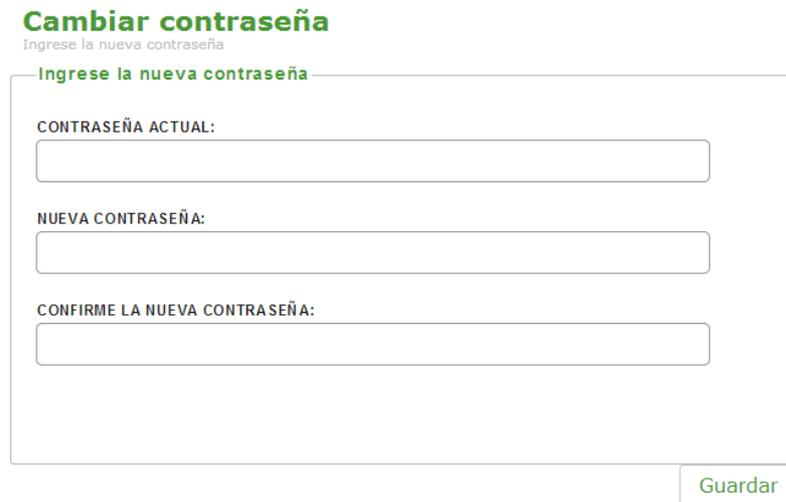


Ilustración 21: Página Web – cambio de contraseña

- Si la contraseña actual es incorrecta, se visualiza el siguiente mensaje de error: “No se ha podido modificar la contraseña, la contraseña actual es incorrecta”. Si el largo de la nueva contraseña es menor a seis caracteres se visualiza el siguiente mensaje de error: “La nueva contraseña debe tener al menos 6 caracteres”.



Ilustración 22: Página Web – mensajes: La contraseña actual es incorrecta, La nueva contraseña debe tener la menos 6 caracteres

- Si la contraseña es modificada correctamente, se visualiza el siguiente mensaje: “La contraseña se ha modificado correctamente”.

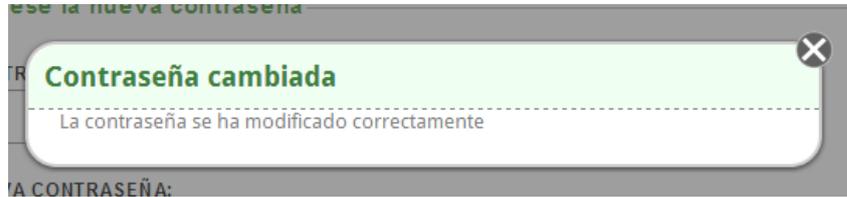


Ilustración 23: Página Web – mensaje: La contraseña se ha modificado correctamente

3.5.2.3. Cerrar Sesión

- Cuando se presiona la opción Cerrar Sesión, se vuelve a la pantalla de inicio.

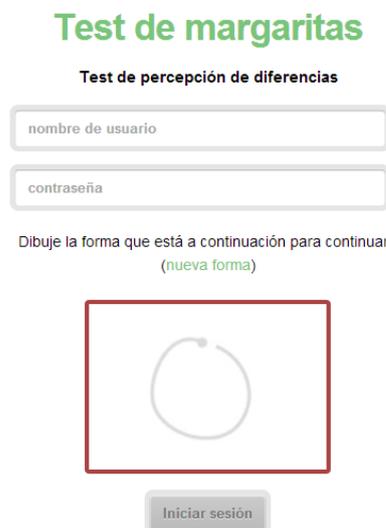


Ilustración 24: Página Web – al seleccionar el botón cerrar sesión, el sistema vuelve a la pantalla de inicio de sesión

3.5.3. Menú Procesos

- Cuando el usuario presiona el menú Procesos, se presentan las siguientes opciones: Agendar Citas, Fichas, Resultados Test.

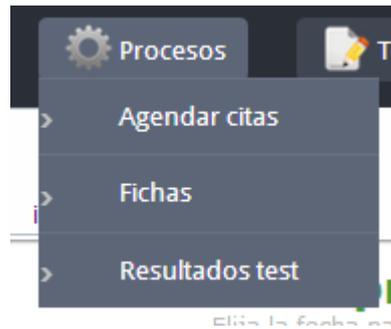


Ilustración 25: Página Web – submenú de procesos

3.5.3.1. Agendar Citas

- En la opción Agendar Citas, se presentan los siguientes botones: Nuevo, Editar, Imprimir y Anular. En la pantalla se visualizan estos botones y también todas las citas programadas para ese día, si se desean ver citas anteriores o posteriores se presiona la fecha y se despliega un calendario para seleccionar el día.

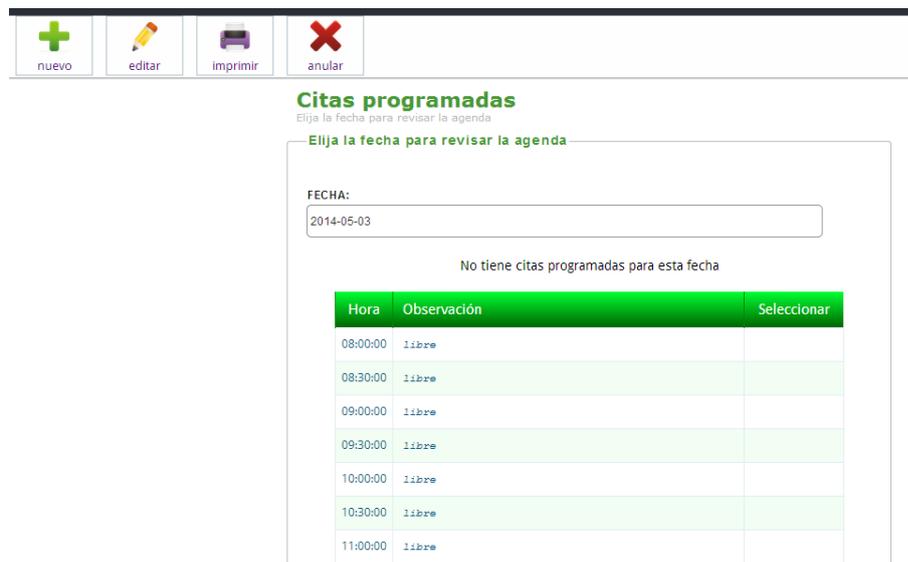


Ilustración 26: Página Web – pantalla de citas programadas

Nuevo

- Se deben ingresar todos los campos obligatorios (*) y el horario de atención, para almacenar la cita se presiona el botón Guardar.

Agendar nueva cita
Ingrese los datos de la nueva cita

Ingrese los datos de la nueva cita

(*)CAMPOS OBLIGATORIOS

FECHA*:
2014-05-03

OBSERVACIÓN*:

ESCOJA LAS HORAS EN LAS QUE DESEA AGENDAR LA CITA*
No tiene citas programadas para esta fecha

Hora	Observación	Seleccionar
08:00:00	libre	<input type="checkbox"/>

Ilustración 27: Página Web – ingreso de una nueva cita

- Si la información almacenada es correcta, se visualiza el siguiente mensaje: “¡Felicitaciones!, la cita se ha agendado correctamente”.

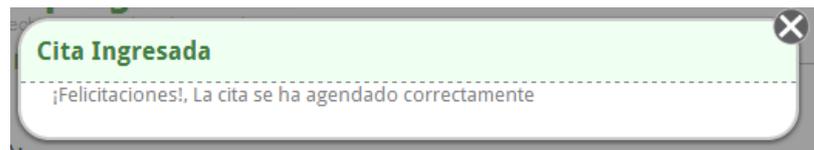


Ilustración 28: Página Web – mensaje: ingreso correcto de la cita

Editar

- El usuario selecciona la cita que deseamos modificar, y presiona el botón Editar.

Citas programadas
Elija la fecha para revisar la agenda

Elija la fecha para revisar la agenda

FECHA*:
2014-05-03

Hora	Observación	Seleccionar
08:00:00	Cita con Christopher Ochoa por bajo rendimiento escolar.	<input type="radio"/>
08:30:00	libre	<input type="radio"/>
09:00:00	Cita con Cristina Espinosa, por episodio depresivo.	<input checked="" type="radio"/>
09:30:00	Cita con Carlos Luis Tello Pardo, por problemas familiares.	<input type="radio"/>
10:00:00	libre	<input type="radio"/>
10:30:00	libre	<input type="radio"/>
11:00:00	Cita con Kathy Torres por bajo rendimiento.	<input type="radio"/>
11:30:00	libre	<input type="radio"/>

Ilustración 29: Página Web – selección de la cita a editar

- Se modifican todos los campos deseados, para almacenarlos se presiona el botón Guardar.

Editar la cita
Ingrese los datos de la cita

Ingrese los datos de la cita

(*)CAMPOS OBLIGATORIOS

FECHA:

2014-05-03

Mayo 2014

Lun	Mar	Mie	Jue	Vie	Sab	Dom
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

UNA A CONTINUACIÓN:*

Hora	Observación	Seleccionar
08:00:00	Cita con Christopher Ochoa por bajo rendimiento escolar.	

Ilustración 30: Página Web – modificación de datos de la cita

- Si la modificación es guardada correctamente, se visualiza el siguiente mensaje: “La cita se ha editado correctamente”.



Ilustración 31: Página Web – mensaje: modificación correcta de la cita

Imprimir

- Si se presiona el botón imprimir, se presenta la siguiente pantalla.

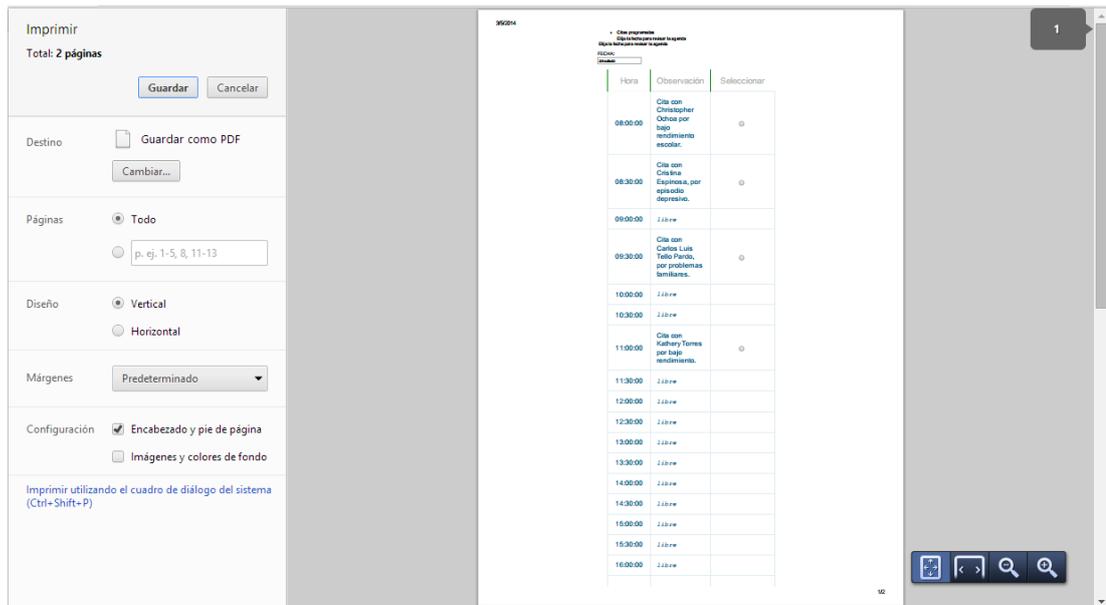


Ilustración 32: Página Web – vista de impresión de citas

Anular

- Si alguna cita fue creada de manera errónea, puede ser anulada. El usuario selecciona la cita y se presiona el botón Anular.



Ilustración 33: Página Web – selección la cita a anular

- El sistema muestra una mensaje de confirmación, si el usuario está seguro de anular esa cita presiona el botón Aceptar, caso contrario presiona Cancelar.

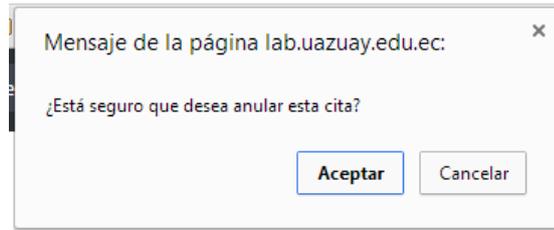


Ilustración 34: Página Web – mensaje de confirmación

- Si la cita fue anulada correctamente se visualiza el siguiente mensaje: “La cita se ha anulado correctamente”.



Ilustración 35: Página Web – mensaje: anulación correcta de la cita

3.5.3.2. Fichas

- La opción Fichas presentan los siguientes botones: Nuevo, Editar, Imprimir, Buscar y Anular. En la pantalla se visualizan estas opciones y un mensaje que indica que no existen fichas ingresadas.



Ilustración 36: Página Web – pantalla de fichas ingresadas

Nuevo

- Para ingresar una nueva ficha se presiona el botón Nuevo, aquí se muestra una pantalla donde se llenan varios campos. Cuando se desee almacenar la ficha, se verifica que todos los campos obligatorios (*) estén llenos y se presiona el botón Guardar.

Ilustración 37: Página Web – ingreso de una nueva ficha

- Si la ficha se guardó correctamente, se visualiza el siguiente mensaje: “¡Felicitaciones!, la ficha se ha ingresado correcta”.

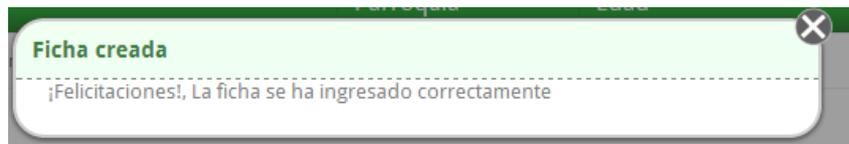


Ilustración 38: Página Web – mensaje: ingreso correcto de la ficha

- Todas las fichas almacenadas se muestran de la siguiente manera.

Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Christopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver - entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Rosa Amada Hurtado López	CUENCA	9 años y 3 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 39: Página Web – lista de todas las fichas ingresadas

Editar

- Para editar una ficha, se presentan dos opciones:
 - En la primera opción, se selecciona la ficha que se va a modificar y luego se presiona el botón editar.



Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver - entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input checked="" type="radio"/>		Rosa Amada Hurtado López	CUENCA	9 años y 3 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 40: Página Web – selección de la ficha a editar, primera opción

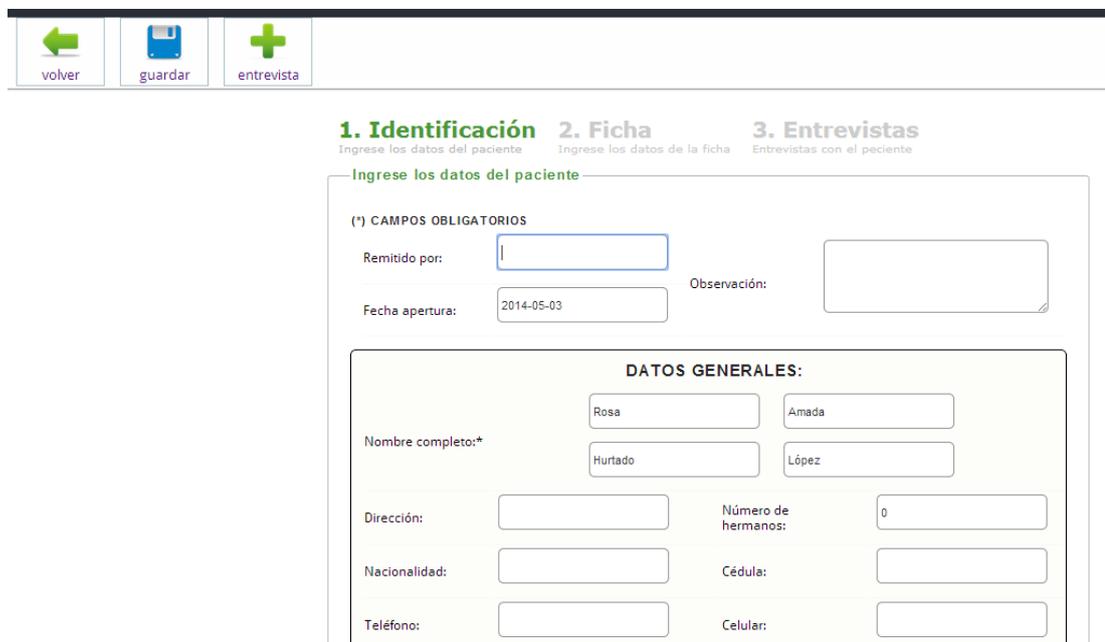
- En la segunda opción, se selecciona la ficha que se va a modificar y luego se presiona el link ver (en la columna Acción).



Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver - entrevista
<input checked="" type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Rosa Amada Hurtado López	CUENCA	9 años y 3 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 41: Página Web – selección de la ficha a editar, segunda opción

- Al seleccionar cualquiera de las dos opciones se muestra la siguiente pantalla, aquí se puede modificar los campos que se requiera. Para almacenar la ficha modificada, se presiona el botón Guardar.



1. Identificación **2. Ficha** **3. Entrevistas**
 Ingrese los datos del paciente Ingrese los datos de la ficha Entrevistas con el paciente

Ingrese los datos del paciente

(*) CAMPOS OBLIGATORIOS

Remitido por:

Fecha apertura:

Observación:

DATOS GENERALES:

Nombre completo:*

Dirección:

Número de hermanos:

Nacionalidad:

Cédula:

Teléfono:

Celular:

Ilustración 42: Página Web – modificación de los datos de la ficha

- Si la ficha modificada es almacenada correctamente, se visualiza el siguiente mensaje: “¡Felicitaciones!, la entrevista se ha ingresado correcta”.

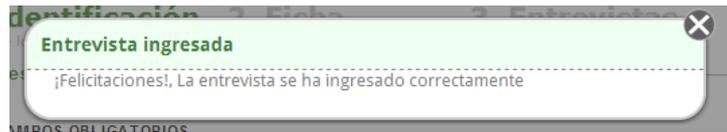


Ilustración 43: Página Web – mensaje: ingreso correcto de la entrevista

- Con la opción de modificar fichas, también se puede agregar entrevistas. Para agregar entrevistas se presentan dos opciones.
 - En la primera opción, se selecciona la ficha deseada y luego se presiona el link entrevista (en la columna Acción).

Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input checked="" type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 44: Página Web – selección de la ficha que se va a entrevistar, primera opción

- Para la segunda opción, cuando se encuentra editando la ficha, se presiona el botón entrevista.

Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input checked="" type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 45: Página Web – selección de la ficha que se va a entrevistar, segunda opción

- Al seleccionar cualquiera de las dos opciones, el sistema permite ingresar una entrevista. Se deben llenar todos los campos necesarios y luego presionar el botón Guardar.

Entrevista
Ingrese los datos de la entrevista

Ingrese los datos de la entrevista

PACIENTE: ROSA AMADA HURTADO LÓPEZ

MOTIVO DE LA CONSULTA
Episodio depresivo

FECHA
2014-05-03

ESTADO PSICOLÓGICO

OBSERVACIÓN

Ilustración 46: Página Web – ingreso de una nueva entrevista

Imprimir

- Para imprimir todas la fichas se presiona el botón Imprimir, el sistema muestra la siguiente pantalla.

Imprimir
Total: 1 página

Guardar Cancelar

Destino Guardar como PDF
Cambiar...

Páginas Todo
 p. ej. 1-5, 8, 11-13

Diseño Vertical
 Horizontal

Márgenes Predeterminado

Configuración Encabezado y pie de página
 Imágenes y colores de fondo

Imprimir utilizando el cuadro de diálogo del sistema (Ctrl+Shift+P)

Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver + entrevista
<input type="radio"/>		Carlos Luis Dolba López	CUENCA	9 años y 8 meses	0	ver + entrevista
<input type="radio"/>		Rosa Amada Hurtado López	CUENCA	9 años y 2 meses	1	ver + entrevista
<input type="radio"/>		Edison Eduardo Viera Castro	MNCAS	5 años y 7 meses	0	ver + entrevista

Ilustración 47: Página Web – vista de impresión de fichas

Buscar

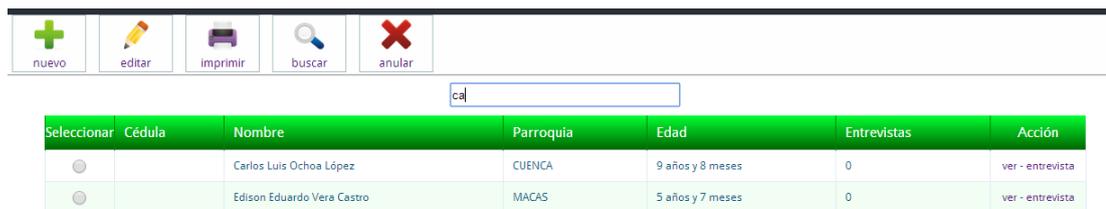
- Si queremos buscar una ficha presionamos el botón buscar, el sistema muestra un cuadro de texto donde se realiza la búsqueda por cédula, nombre o parroquia del sujeto evaluación.



Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver - entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Rosa Amada Hurtado López	CUENCA	9 años y 3 meses	1	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 48: Página Web – selección del botón buscar

- Si se ingresan dos caracteres como mínimo se filtra la información y se muestra únicamente los caracteres buscados.



Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 49: Página Web – ingreso de caracteres a buscar

Anular

- Si alguna ficha fue creada de forma errónea puede ser anulada. Se selecciona la ficha que deseamos y presionamos el botón Anular.



Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver - entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input checked="" type="radio"/>		Rosa Amada Hurtado López	CUENCA	9 años y 3 meses	1	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 50: Página Web – selección de la ficha a anular

- El sistema muestra un mensaje de confirmación, si estamos seguros de anular esa ficha presionamos el botón Aceptar, caso contrario presionamos Cancelar.

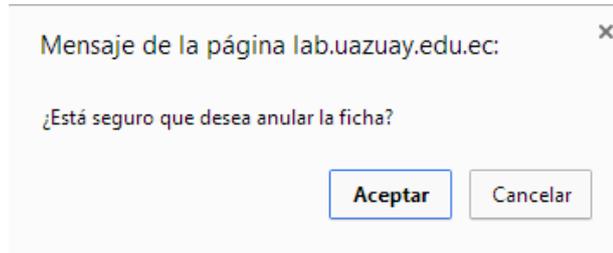


Ilustración 51: Página Web – mensaje de confirmación

- Si la ficha fue anulada correctamente, el sistema visualiza el siguiente mensaje: “La ficha se ha anulado correctamente”.



Ilustración 52: Página Web – mensaje: anulación correcta de la ficha

- Cuando listemos todas las fichas, la ficha anulada ya no se visualiza.

Seleccionar	Cédula	Nombre	Parroquia	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	CUENCA	8 años y 2 meses	0	ver - entrevista
<input type="radio"/>		Carlos Luis Ochoa López	CUENCA	9 años y 8 meses	0	ver - entrevista
<input type="radio"/>		Edison Eduardo Vera Castro	MACAS	5 años y 7 meses	0	ver - entrevista

Ilustración 53: Página Web – lista de todas las fichas existentes

3.5.3.3. Resultados Test

- Con la opción Resultados Test el sistema visualiza todas las fichas ingresadas, además el usuario puede buscar las fichas por cédula y nombre. Al presionar en la Acción ver resultados, se visualiza todos test realizados por el sujeto evaluación.

Resultados de aplicaciones de tests

Elija al sujeto de evaluación

Elija al sujeto de evaluación

BUSCAR:

Parroquia	Cédula	Nombre	Edad	Entrevistas	Acción
CUENCA		Cristopher Julian Zhunio Ochoa	8 años y 2 meses	0	ver resultados
CUENCA		Carlos Luis Ochoa López	9 años y 8 meses	0	ver resultados
MACAS		Edison Eduardo Vera Castro	5 años y 7 meses	0	ver resultados

Ilustración 54: Página Web – lista de todas las fichas existentes, selección de resultados

- El sistema visualiza el nombre, tiempo, comisión, aciertos, errores, omisiones, dispositivo y acción, de todos los test realizados por el sujeto evaluación.



volver



imprimir

Nombre: Christopher Julian Zhunio Ochoa
Edad: 8 años y 2 meses

Fecha(toma del test)	Nombre del Test	Tiempo (mins)	Comisión	Aciertos	Errores	Omisiones	Dispositivo	Acción
2014-05-03 23:17:00	Test Triple Margarita - Interferencia	0.611	0	18	12	0	WEB	ver detalle
2014-05-03 23:16:15	Test Margarita	1.006	3	19	11	0	WEB	ver detalle
2014-05-03 23:11:15	Test Margarita - Interferencia	2.297	3	16	9	5	WEB	ver detalle

Ilustración 55: Página Web – lista de todos los test tomados a la ficha seleccionada

- Si el usuario presiona el botón Imprimir, se presenta la siguiente pantalla.

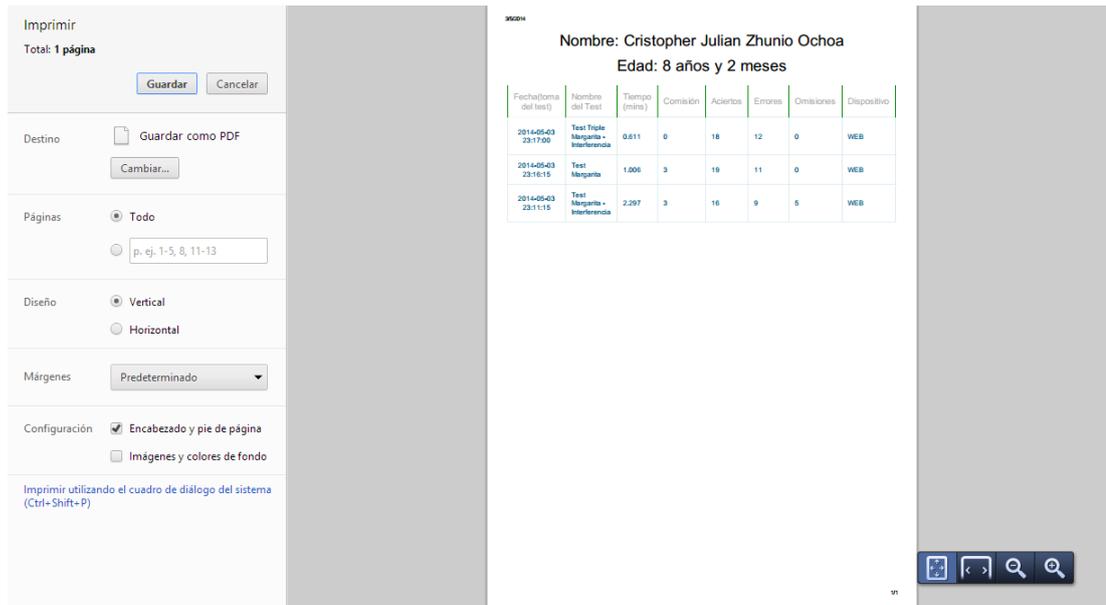


Ilustración 56: Página Web – vista de impresión de los test tomados

- Si el usuario presiona en la acción Ver Detalle, el sistema muestra de forma detallada los aciertos, errores, omisiones y el tiempo en responder, de cada pregunta del test y de las interferencias.

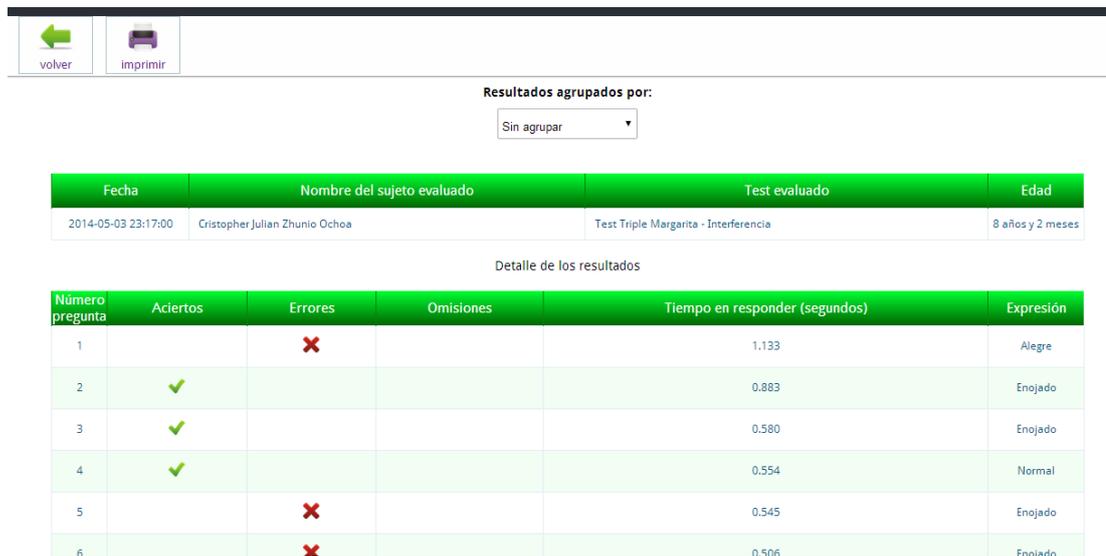


Ilustración 57: Página Web – detalle del test seleccionado

- Cuando el usuario presiona el botón Imprimir, se muestra la siguiente pantalla.

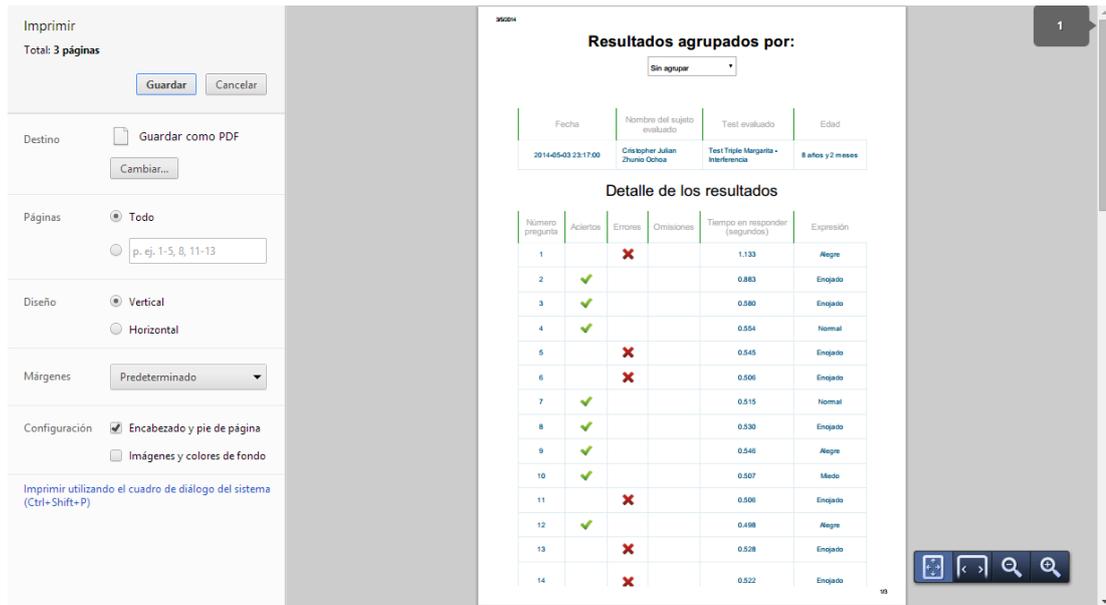


Ilustración 58: Página Web – vista de impresión del test seleccionado

- Si el usuario presiona la opción Agrupar por expresión, los resultados se agrupan por: alegre, enojado, normal, miedo, triste.



Ilustración 59: Página Web – selección de resultados agrupar por expresión

3.5.4. Menú Tomar Test

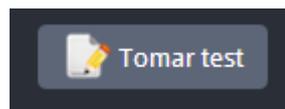


Ilustración 60: Página Web – menú tomar test

- Cuando se presiona el menú Tomar Test, el sistema visualiza todas las fichas creadas. Además muestra la opción de buscar fichas por cédula o nombre.

1. Sujeto de evaluación

Elija al sujeto de evaluación

2. Test

Elija el test que se va a evaluar

Elija al sujeto de evaluación

BUSCAR:

Seleccionar	Cédula	Nombre	Edad	Entrevistas	Acción
<input type="radio"/>		Cristopher Julian Zhunio Ochoa	8 años y 2 meses	0	ver ficha
<input type="radio"/>		Carlos Luis Ochoa López	9 años y 8 meses	0	ver ficha
<input type="radio"/>		Edison Eduardo Vera Castro	5 años y 7 meses	0	ver ficha

[Continuar >](#)

Ilustración 61: Página Web – lista de todas las fichas existentes, ver ficha

- Para mostrar la ficha completa del sujeto evaluación, se presiona en la opción ver ficha y el sistema visualiza toda la información del sujeto seleccionado. Presione en el botón Volver para regresar a la pantalla anterior.

[← volver](#)

DATOS GENERALES:

Nombre completo:*

Dirección: Número de hermanos:

Nacionalidad: Cédula:

Teléfono: Celular:

E-mail: Sexo:*

Lugar de nacimiento: Fecha nacimiento:*

Edad:

DATOS ESCUELA:

PROVINCIA:* CANTÓN:* PARROQUIA:*

Escuela: Nivel escolar:

Ilustración 62: Página Web – vista de la información de la ficha seleccionada

- Para tomar un test, seleccione el nombre del sujeto evaluación y presione el botón Continuar.



Ilustración 63: Página Web – selección de la ficha a tomar el test

- El sistema muestra todos los test creados hasta el momento, junto con la descripción, fecha de creación, fecha de modificación y las interferencias de cada uno.

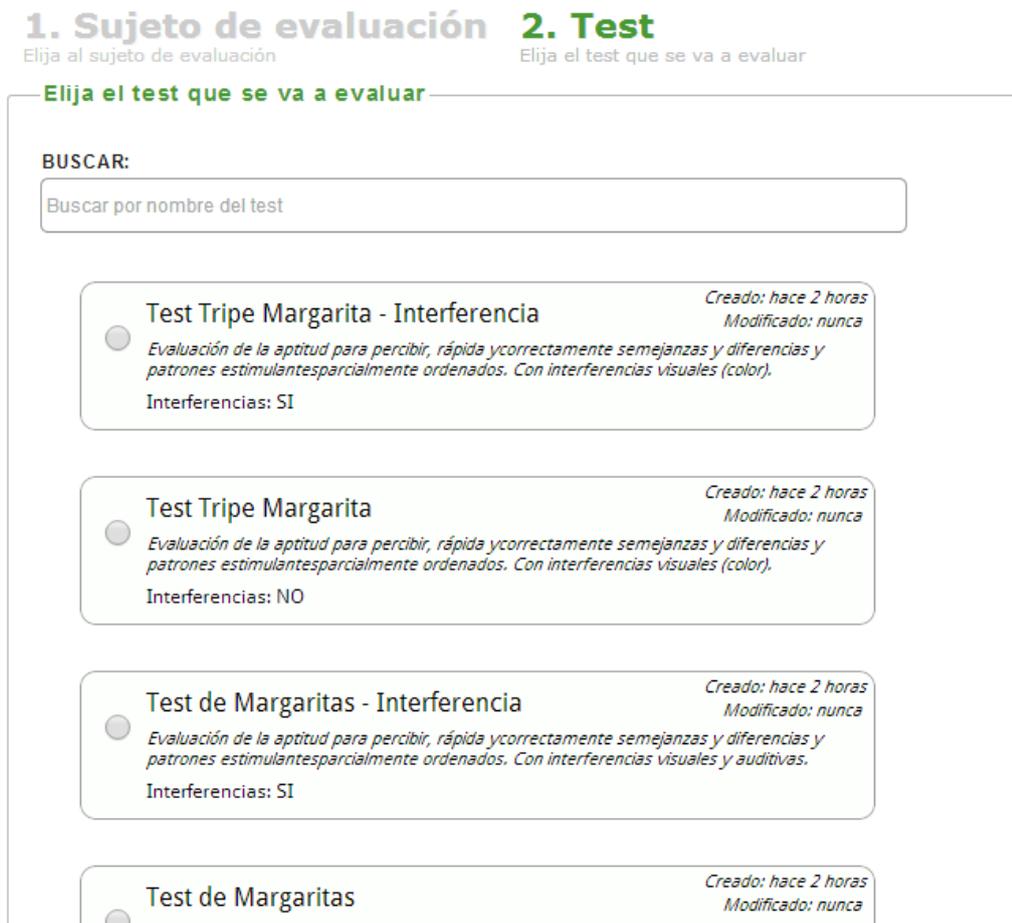


Ilustración 64: Página Web – lista de todos los test existentes

- El usuario puede buscar los test por su nombre. La información ingresada se filtra y se muestran los test que contengan a los caracteres digitados.

1. Sujeto de evaluación

Elija al sujeto de evaluación

2. Test

Elija el test que se va a evaluar

Elija el test que se va a evaluar

BUSCAR:

Test Tripe Margarita - Interferencia Creado: hace 2 horas
Modificado: nunca

Evaluación de la aptitud para percibir, rápida y correctamente semejanzas y diferencias y patrones estimulantes parcialmente ordenados. Con interferencias visuales (color).

Interferencias: SI

Test de Margaritas - Interferencia Creado: hace 2 horas
Modificado: nunca

Evaluación de la aptitud para percibir, rápida y correctamente semejanzas y diferencias y patrones estimulantes parcialmente ordenados. Con interferencias visuales y auditivas.

Interferencias: SI

Ilustración 65: Página Web – ingreso de caracteres a buscar

- Para tomar el test, el usuario selecciona el test que se va a aplicar al sujeto evaluación y presiona el botón Empezar Test.

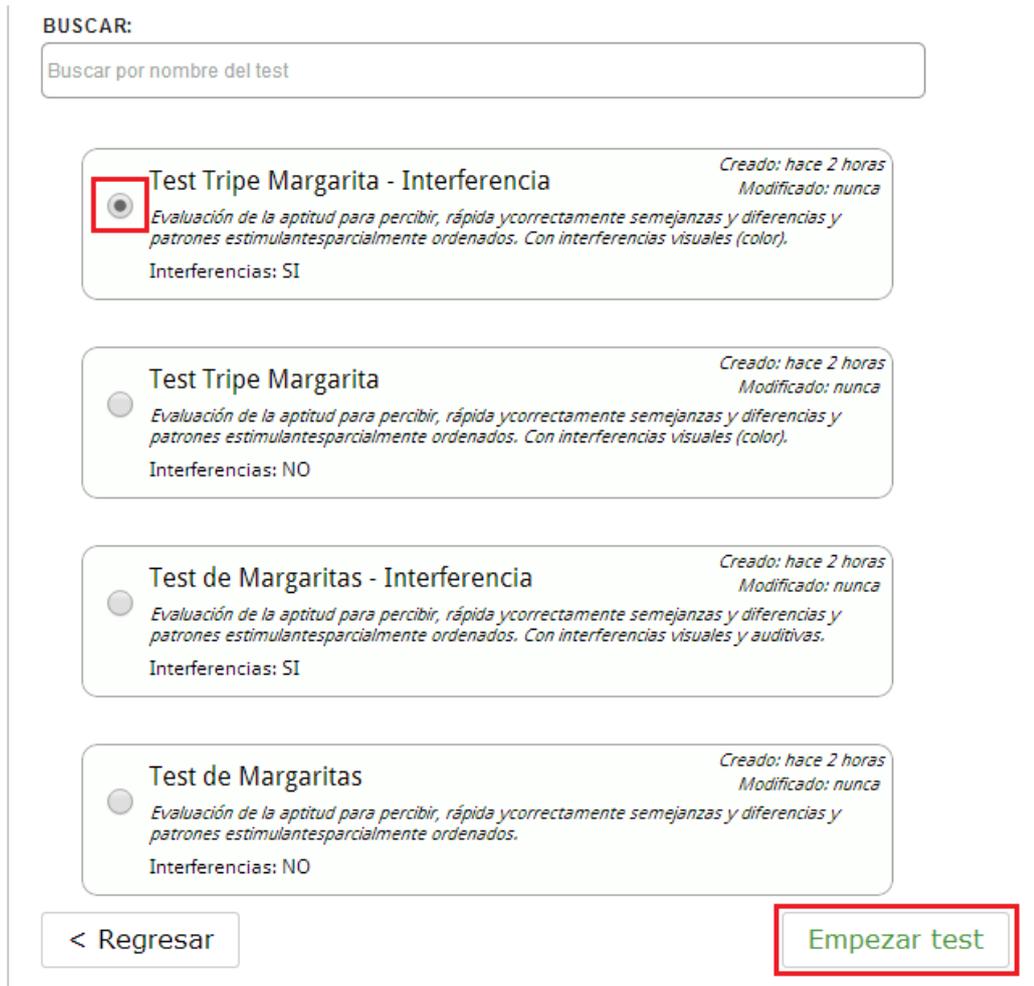


Ilustración 66: Página Web – selección del test que se va a evaluar

- Si la los pasos son correctos, se muestran las instrucciones y el test.

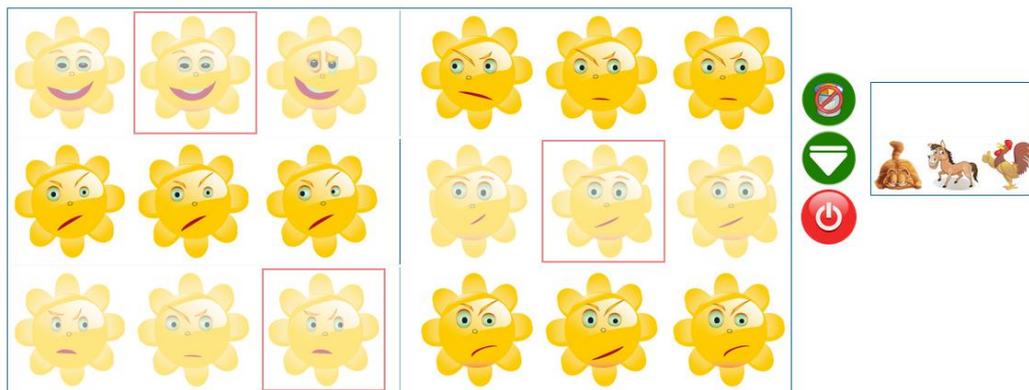


Ilustración 67: Página Web – vista del test selccionado

- Cuando el sujeto evaluación termina correctamente el test el sistema visualiza la siguiente pantalla, presionamos el botón Aceptar.

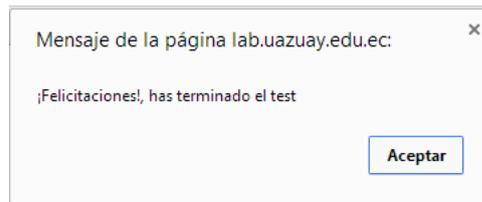


Ilustración 68: Página Web – mensaje de información

- El sistema visualiza el siguiente mensaje: “La aplicación de test se ha ingresado correctamente”, para confirmar que el test se ha guardado correctamente.

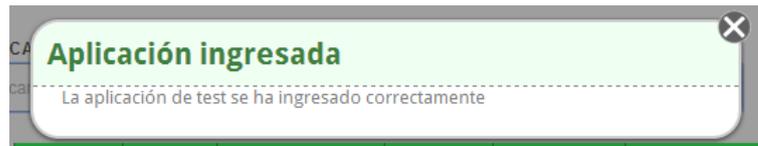


Ilustración 69: Página Web – mensaje: ingreso correcto del test

3.5.5. Menú Acerca de

- Cuando el usuario selecciona el menú Acerca de, se visualiza toda la información sobre el desarrollo del test.



Ilustración 70: Página Web – vista del menú acerca de

3.5.6. Menú Administración

- El menú Administración, únicamente es visible para los usuarios administradores, en usuarios normales esta opción no se visualiza. Aquí se presenta las siguiente opciones: Motivo de consulta, Administrar test, Administrar usuario, Administrar escuelas, Subir margaritas.

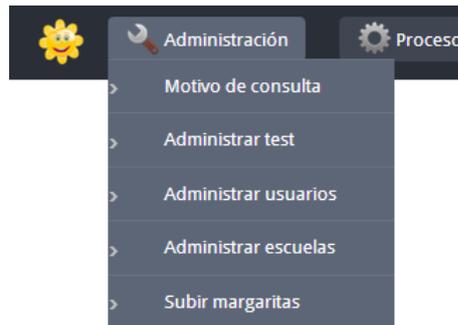


Ilustración 71: Página Web – submenú administración

3.5.6.1. Motivo de consulta

- En la opción Motivo de consulta se presentan los siguientes botones: Nuevo, Editar, Imprimir, Buscar y Anular. En la pantalla se visualizan estos botones y un mensaje que indica que no existen motivos de consulta ingresados.



Ilustración 72: Página Web – pantalla de motivos consulta existentes

Nuevo

- Para ingresar un nuevo Motivo de Consulta presionamos el botón Nuevo, aquí se muestra una pantalla donde debemos llenar el campo. Si el usuario quiere almacenar la información, presiona el botón guardar.



Ilustración 73: Página Web – ingreso de un nuevo motivo consulta

- Si el motivo de consulta se guardó correctamente, se visualiza el siguiente mensaje: “¡Felicitaciones!, el motivo de consulta se ha ingresado correcta”.

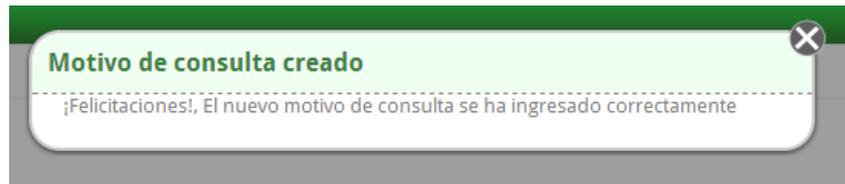


Ilustración 74: Página Web – mensaje: ingreso correcto del motivo consulta

- Todos los motivos de consulta almacenados se muestran de la siguiente manera.

Seleccionar	Motivo de la Consulta	Código
<input type="radio"/>	Episodio depresivo	1
<input type="radio"/>	Bajo rendimiento	2
<input type="radio"/>	Problemas familiares	3
<input type="radio"/>	Trastorno de sueño	4
<input type="radio"/>	Crisis de angustia	5

Ilustración 75: Página Web – lista de todos los motivos consulta ingresados

Editar

- Para modificar el motivo de consulta, seleccionamos la opción que se va a modificar y luego presionamos el botón editar.

Seleccionar	Motivo de la Consulta	Código
<input type="radio"/>	Episodio depresivo	1
<input type="radio"/>	Bajo rendimiento	2
<input checked="" type="radio"/>	Problemas familiares	3
<input type="radio"/>	Trastorno de sueño	4
<input type="radio"/>	Crisis de angustia	5

Ilustración 76: Página Web – selección del motivo consulta a editar

- El sistema visualiza la siguiente pantalla, aquí podemos modificar el motivo consulta. Para almacenar la información se presiona el botón Guardar.

Ilustración 77: Página Web – modificación de los datos del motivo consulta

- Si el Motivo Consulta se modificó correctamente, se visualiza el siguiente mensaje: “El motivo de consulta se ha modificado correctamente”.

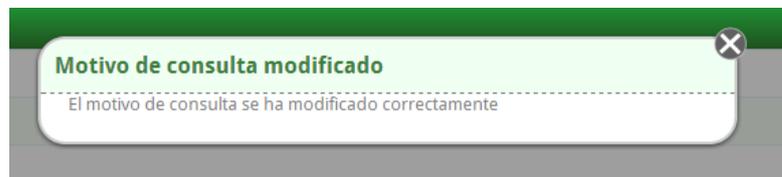


Ilustración 78: Página Web – mensaje: modificación correcta del motivo consulta

Imprimir

- Si usuario presiona el botón Imprimir, el sistema muestra la siguiente pantalla.

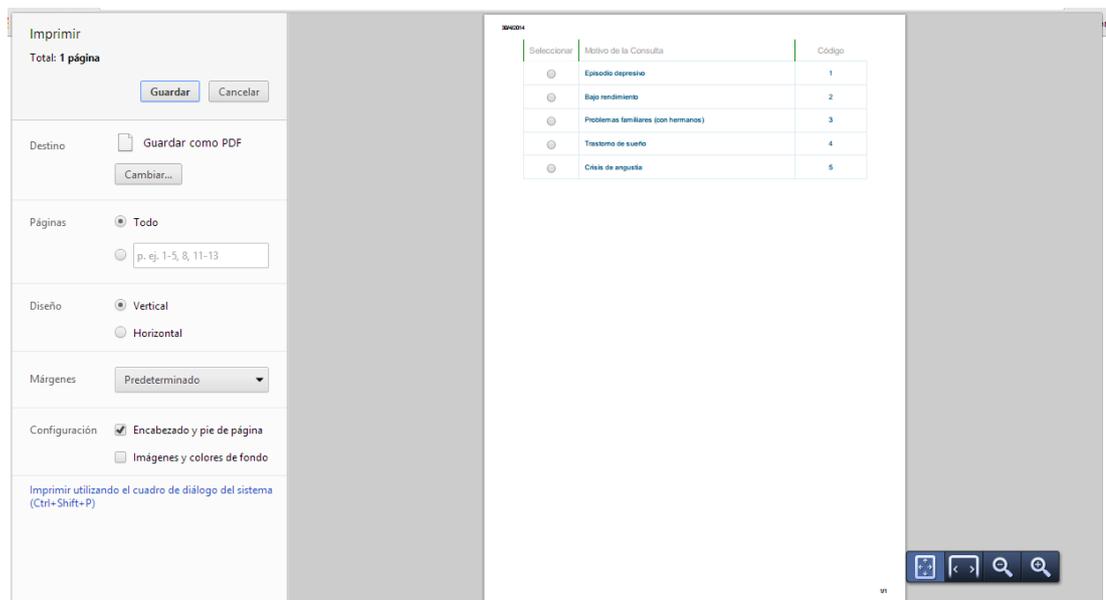


Ilustración 79: Página Web – vista de impresión de motivo consulta

Buscar

- Si el usuario presiona el botón Buscar, el sistema muestra un cuadro de texto donde se digita caracteres para realizar la búsqueda por el motivo de consulta.



Ilustración 80: Página Web –ingreso de caracteres a buscar

Anular

- Si algún motivo de consulta fue creado de forma errónea puede ser anulado. Se selecciona la ficha y presiona el botón Anular.

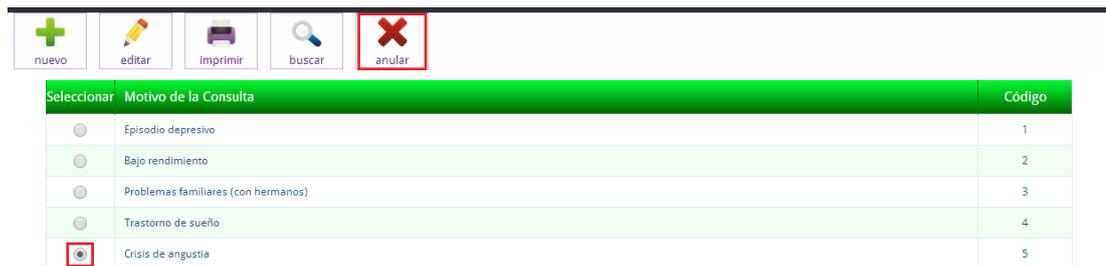


Ilustración 81: Página Web – selección del motivo consulta a anular

- El sistema muestra un mensaje de confirmación, si estamos seguros de anular el motivo consulta presionamos el botón Aceptar, caso contrario Cancelar.

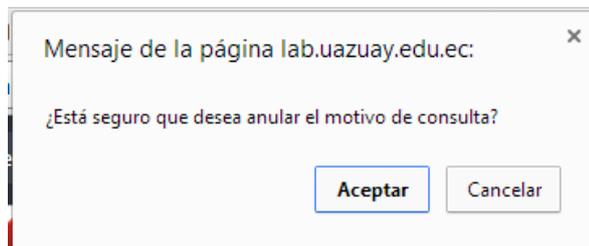


Ilustración 82: Página Web – mensaje de confirmación

- Si el motivo consulta fue anulado correctamente, el sistema visualiza el siguiente mensaje: “El motivo consulta se ha anulado correctamente”.

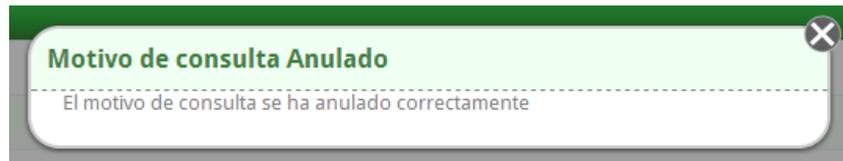


Ilustración 83: Página Web – mensaje: anulación correcta del motivo consulta

- Cuando listemos todos los motivos consultas, el motivo anulado ya no se visualiza.

Seleccionar	Motivo de la Consulta	Código
<input type="radio"/>	Episodio depresivo	1
<input type="radio"/>	Bajo rendimiento	2
<input type="radio"/>	Problemas familiares (con hermanos)	3
<input type="radio"/>	Trastorno de sueño	4

Ilustración 84: Página Web – lista de todos los motivos consulta existentes

3.5.6.2. Administrar test

- La opción Administrar Test, se presentan los siguientes botones: Nuevo, Editar, Imprimir, Buscar y Anular. En la pantalla se visualizan estos botones y un mensaje que indica que no existen test ingresados.



Ilustración 85: Página Web – pantalla de test ingresados

Nuevo

- Para ingresar un nuevo test se presiona botón Nuevo, el sistema muestra una pantalla donde el obligatorio llenan todos los campos.


 volver


 guardar

1. Datos Generales

Datos generales del nuevo test

2. Elegir Margaritas

Elija el orden y las margaritas del nuevo test

Datos generales del nuevo test

NOMBRE:

DESCRIPCIÓN:

NÚMERO DE FILAS DE MARGARITAS POR PANTALLA:

¿TIENE INTERFERENCIAS?:

Ilustración 86: Página Web – ingreso de un nuevo test – datos generales

- A continuación el usuario debe presionar la opción Elegir Margaritas, el sistema visualiza todas las margaritas que se hayan subido con anterioridad. El usuario selecciona las margaritas que desee, en este caso, mínimo una y máximo treinta.

1. Datos Generales

Datos generales del nuevo test

2. Elegir Margaritas

Elija el orden y las margaritas del nuevo test

Elija el orden y las margaritas del nuevo test

Seleccionar todos
Ha seleccionado 30 series

↑	<input checked="" type="checkbox"/>				CORRECTO 3	EXPRE SIÓN Alegre
↑	<input checked="" type="checkbox"/>				CORRECTO 3	EXPRE SIÓN Enojado
↑	<input checked="" type="checkbox"/>				CORRECTO 2	EXPRE SIÓN Enojado
↑	<input checked="" type="checkbox"/>				CORRECTO 1	EXPRE SIÓN Normal

Ilustración 87: Página Web – ingreso de un nuevo test – selección de margaritas

- Con el ratón el usuario puede mover las margaritas a la posición que desee, es decir, si la margarita correcta está en la posición tres, puede ser movida a la posición uno. Además puede modificar que margarita es correcta (1, 2, 3) y la Expresión (alegre, enojado, normal, miedo, triste).



Ilustración 88: Página Web – ingreso de un nuevo test – cambio del orden de las margaritas

- Para almacenar la información, el usuario presiona el botón guardar.



Ilustración 89: Página Web – ingreso de un nuevo test – guardar las margaritas seleccionadas

- Si el test se guardó correctamente, se visualiza el siguiente mensaje: “¡Felicitaciones!, el test se ha creado correcta”.

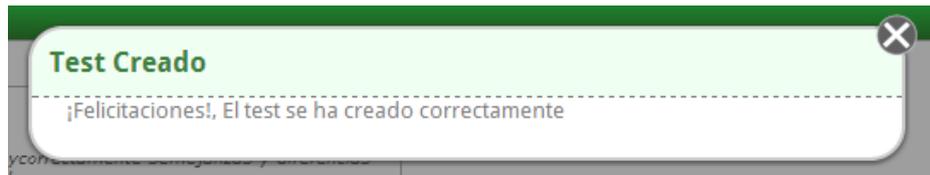


Ilustración 90: Página Web – mensaje: ingreso correcto del test

- Todos los test almacenados se visualizan de la siguiente manera.

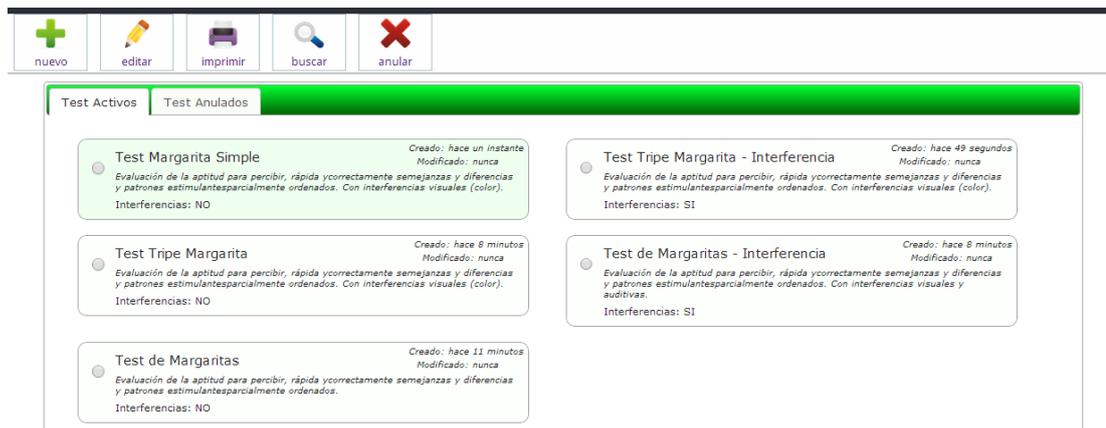


Ilustración 91: Página Web – lista de todos los test ingresados

Editar

- Para modificar el test, el usuario selecciona el test que se va a modificar y luego presiona en el botón editar.

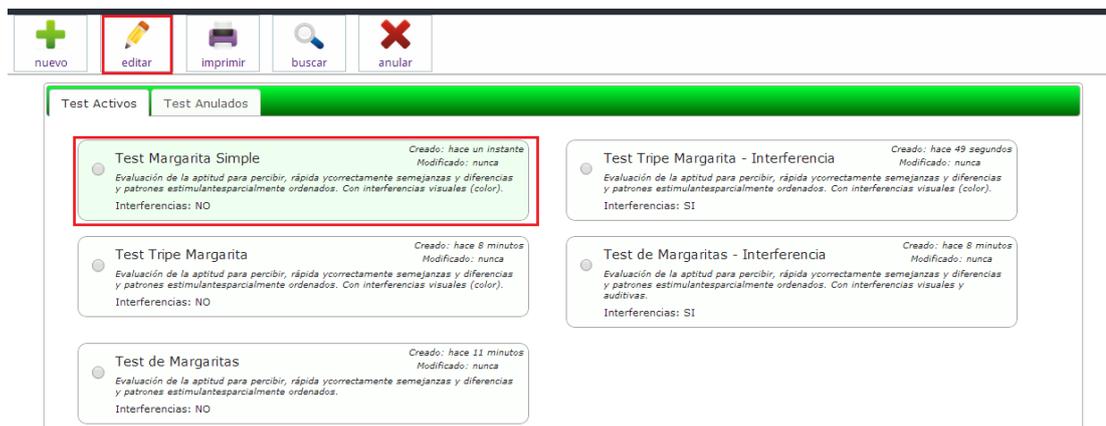
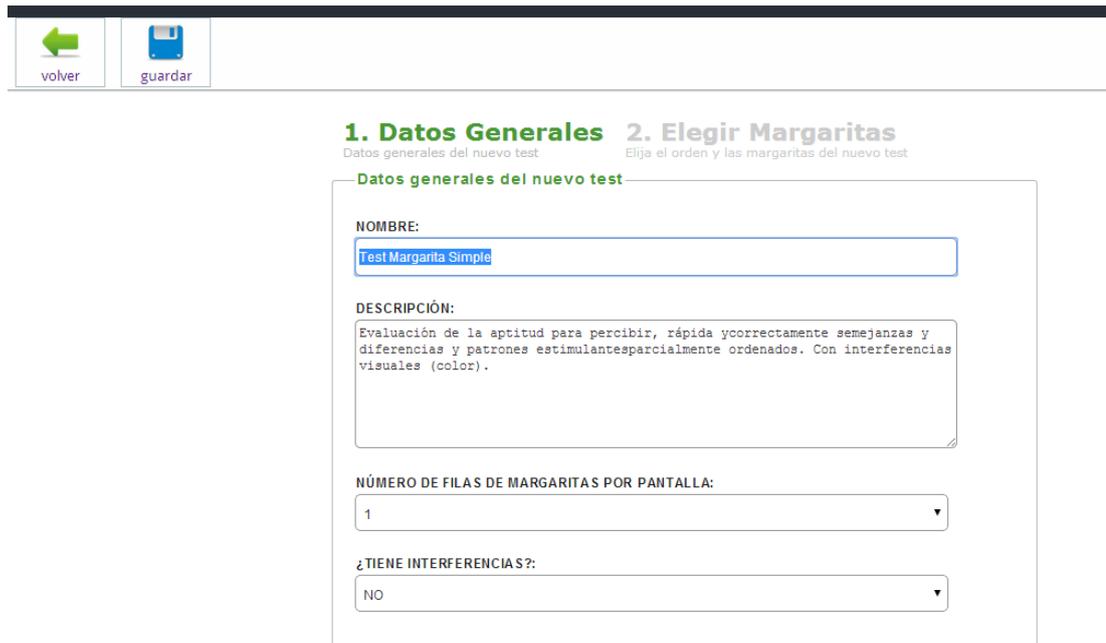


Ilustración 92: Página Web – selección del test a editar

- El sistema visualiza la siguiente pantalla, en donde se puede modificar los datos del test. Cuando se quiere almacenar la modificación del test, se presiona el botón Guardar.



1. Datos Generales **2. Elegir Margaritas**
Datos generales del nuevo test Elija el orden y las margaritas del nuevo test

Datos generales del nuevo test

NOMBRE:
Test Margarita Simple

DESCRIPCIÓN:
Evaluación de la aptitud para percibir, rápida y correctamente semejanzas y diferencias y patrones estimulantes parcialmente ordenados. Con interferencias visuales (color).

NÚMERO DE FILAS DE MARGARITAS POR PANTALLA:
1

¿TIENE INTERFERENCIAS?:
NO

Ilustración 93: Página Web – modificación de datos del test

- Si el test se modificó correctamente, se visualiza el siguiente mensaje: “¡Felicitaciones!, El test se ha modificado correctamente”.

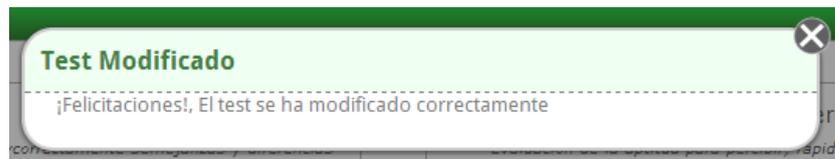


Ilustración 94: Página Web – mensaje: modificación correcta del test

Imprimir

- Si el usuario presiona el botón Imprimir, el sistema muestra la siguiente pantalla.

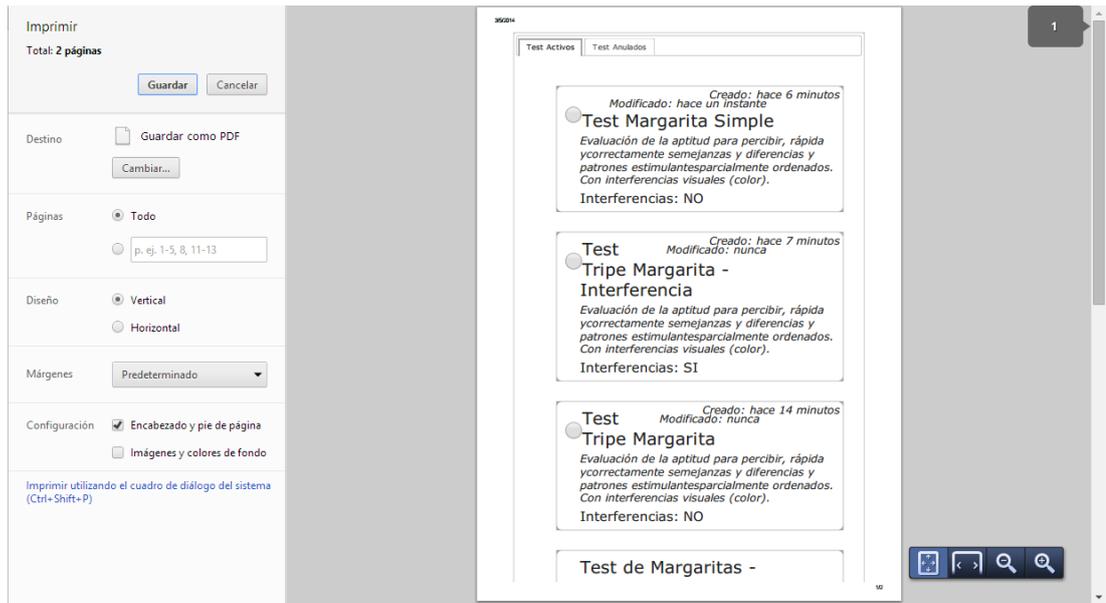


Ilustración 95: Página Web – Vista de la impresión de los test

Buscar

- Si el usuario presiona el botón Buscar, el sistema muestra un cuadro de texto donde podemos realizar la búsqueda por el nombre del test.

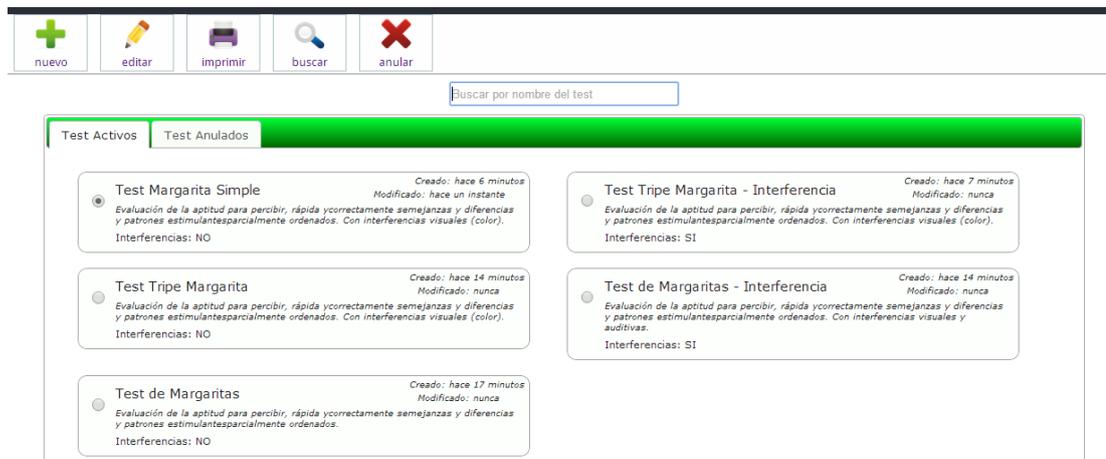


Ilustración 96: Página Web – selección e ingreso de los caracteres a buscar

Anular

- Si algún test fue creado de forma errónea puede ser anulado. Se selecciona un test y se presiona el botón Anular.

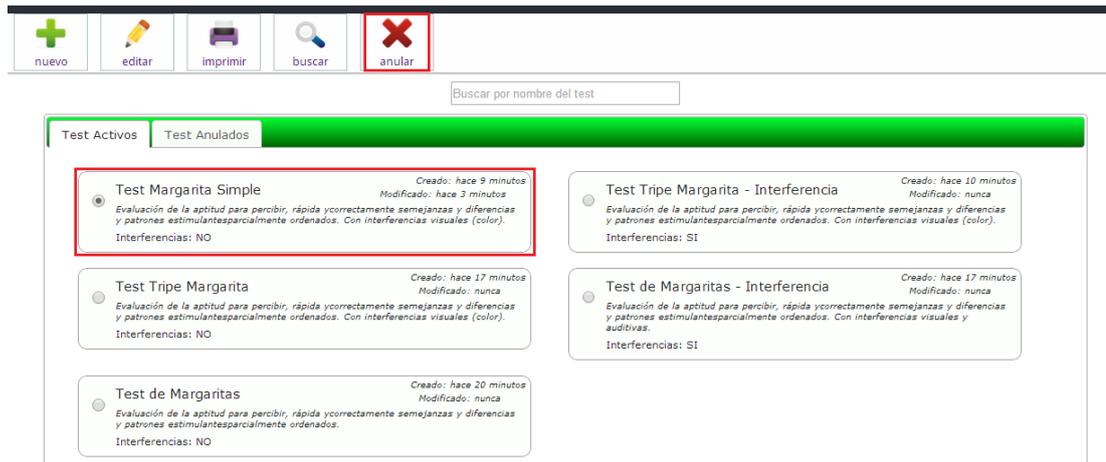


Ilustración 97: Página Web – selección del test a anular

- El sistema muestra un mensaje de confirmación, si estamos seguros de anular el test presionamos el botón Aceptar, caso contrario Cancelar.

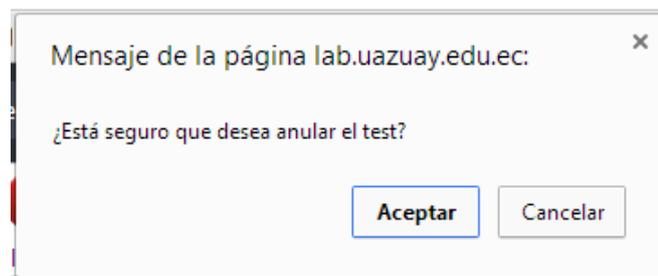


Ilustración 98: Página Web – mensaje de confirmación

- Si el test fue anulado correctamente, el sistema visualiza el siguiente mensaje: “El test se ha anulado correctamente”.

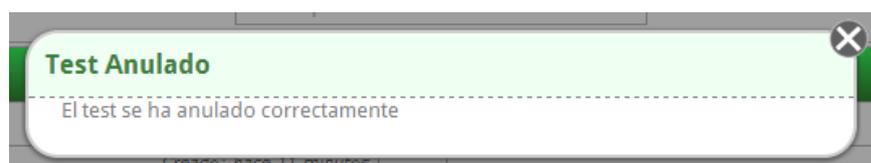


Ilustración 99: Página Web – mensaje: anulación correcta del test

- Cuando el usuario presiona la opción Test Anulados, el sistema visualiza todos los test que fueron anulados con anterioridad, se puede seleccionar la opción activar para volver a utilizar dicho test.



Ilustración 100: Página Web – lista de todos los test anulados

3.5.6.3. Administrar usuarios

- En la opción Administrar usuarios se presentan los siguientes botones: Nuevo, Editar, Buscar y Anular. En la pantalla se visualizan estos botones y todos los usuarios creados anteriormente.



Ilustración 101: Página Web – pantalla de usuarios existentes

Nuevo

- Para ingresar un nuevo usuario presionamos en la botón Nuevo, aquí se muestra una pantalla donde se ingresan los campos obligatorios (*). Para almacenar la información, presionamos el botón guardar.

Ilustración 102: Página Web – ingreso de un nuevo usuario

- El número de cédula es único, si se digita un número de cédula repetido el sistema lanza el siguiente error: “Esta cédula ya existe, ingrese otra”.

Ingrese los datos del nuevo usuario

(*) CAMPOS OBLIGATORIOS

CÉDULA:*

Esta cédula ya existe, ingrese otra

PRIMER NOMBRE:*

Ilustración 103: Página Web – mensaje: cédula ya existente

- El nombre de usuario es un campo único, si se digita un nombre de usuario ya existente el sistema lanza el siguiente error: “Este usuario ya existe, ingrese otro”.

NOMBRE DE USUARIO:*

Este usuario ya existe, ingrese otro

CONTRASEÑA:*

Ilustración 104: Página Web – mensaje: usuario ya existente

- Si el usuario se creó correctamente, se visualiza el siguiente mensaje: “¡Felicitaciones!, el usuario se ha ingresado correctamente”.

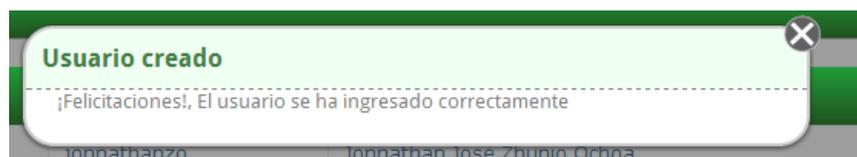


Ilustración 105: Página Web – mensaje: ingreso correcto del usuario

- Todos los usuarios almacenados se muestran de la siguiente manera.

Usuarios Activos		Usuarios Anulados		
Seleccionar	Cédula	Usuario	Nombre	E-mail
<input type="radio"/>	0105016448	jonnathanzo	Jonnathan José Zhunio Ochoa	jonnathanz8a@me.com
<input type="radio"/>	111111111	jennifer	Jennifer Cecibel Torres Hurtado	
<input type="radio"/>	172350594	carlos_luis	Carlos Luis Ochoa Cabrera	carlos_luis@hotmail.com
<input type="radio"/>	0101995223	mcobos	Martha Esperanza Cobos Cali	mcobos@uazuay.edu.ec
<input type="radio"/>	0103379525	castudillo	Catalina Astudillo	cvastudillo@uazuay.edu.ec

Ilustración 106: Página Web – lista de todos los test ingresados

Editar

- Para modificar los datos del usuario, seleccionamos el usuario que se va a modificar y luego presionamos el botón editar.



Ilustración 107: Página Web – selección del usuario a editar

- El sistema visualiza la siguiente pantalla donde se modifican los datos que se crean necesarios. Para almacenar la modificación del usuario, presionamos el botón Guardar.

Datos Usuario
Ingrese los datos del usuario

Ingrese los datos del usuario

(*) CAMPOS OBLIGATORIOS
CÉDULA: 1723550594

PRIMER NOMBRE:*

SEGUNDO NOMBRE:

APELLIDO PATERNO:*

APELLIDO MATERNO:

E-MAIL:

Ilustración 108: Página Web – Modificación de los datos del usuario

- Si el usuario se modificó correctamente, se visualiza el siguiente mensaje: “El usuario se ha modificado correctamente”.

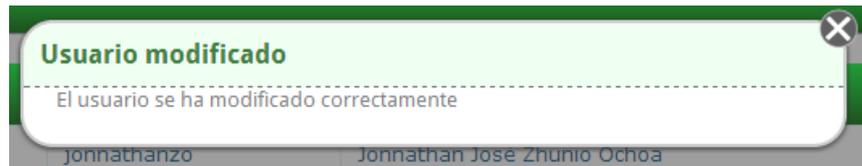


Ilustración 109: Página Web – mensaje: modificación correcta del usuario

Buscar

- Si el usuario presiona el botón Buscar, el sistema muestra un cuadro de texto donde podemos realizar la búsqueda por cédula, usuario o nombre.



Ilustración 110: Página Web – selección e ingreso de los caracteres a buscar

Anular

- Si algún usuario fue creado de forma errónea puede ser anulado. Se selecciona el usuario que deseamos y presionamos el botón Anular.

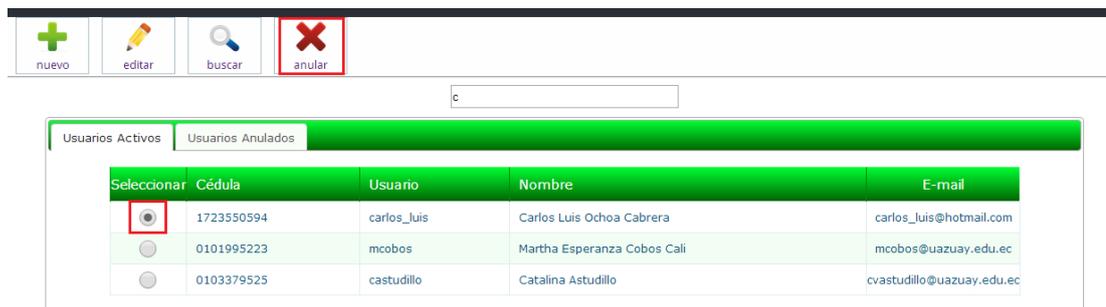


Ilustración 111: Página Web – selección del usuario a anular

- El sistema muestra un mensaje de confirmación, si está seguro de anular el usuario presiona el botón Aceptar, caso contrario Cancelar.

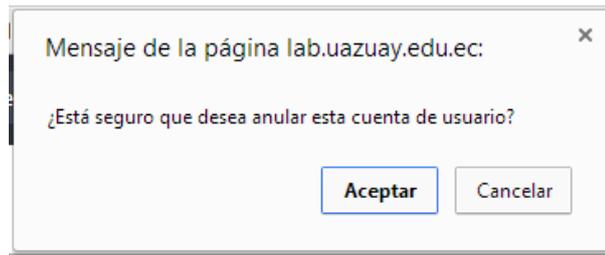


Ilustración 112: Página Web – mensaje de confirmación

- Si el usuario fue anulado correctamente, el sistema visualiza el siguiente mensaje: “El usuario se ha anulado correctamente”.



Ilustración 113: Página Web – mensaje: anulación correcta del usuario

- Cuando se listan todos los usuarios, el usuario anulado ya no se visualiza.

Usuarios Activos		Usuarios Anulados			
Seleccionar	Cédula	Usuario	Nombre	E-mail	
<input type="radio"/>	0105016448	jonnathanzo	Jonnathan José Zhunio Ochoa	jonnathanz8a@me.com	
<input type="radio"/>	111111111	jennifer	Jennifer Cecibel Torres Hurtado		
<input type="radio"/>	0101995223	mcobos	Martha Esperanza Cobos Cali	mcobos@uazuay.edu.ec	
<input type="radio"/>	0103379525	castudillo	Catalina Astudillo	cvastudillo@uazuay.edu.ec	

Ilustración 114: Página Web – lista de todos los usuarios existentes

- Si el usuario selecciona la opción Usuario Anulados, se visualizan todos los usuarios anulados previamente, para que vuelva a estar habilitados presionamos la acción activar.

Usuarios Activos		Usuarios Anulados				
Seleccionar	Cédula	Usuario	Nombre	E-mail	Acción	
<input type="radio"/>	1723550594	carlos_luis	Carlos Luis Ochoa Cabrera	carlos_luis@hotmail.com	activar	

Ilustración 115: Página Web – lista de todos los usuarios anulados

3.5.6.4. Administrar escuelas

- En la opción Administrar Escuelas se visualizan los siguientes botones: Nuevo, Anular. Además se presenta un mensaje que indica que no existen escuelas ingresadas.



Ilustración 116: Página Web – pantalla de escuelas ingresadas

Nuevo

- Si el usuario presiona el botón Nuevo, el sistema muestra una pantalla donde se deben llenar todos los campos obligatorios (*). Para almacenar la información necesaria el usuario presiona el botón Guardar.

Ilustración 117: Página Web –ingreso de una nueva escuela

- Si los datos se almacenan de forma correcta, el sistema lanza el siguiente mensaje: “¡Felicitaciones!, la escuela se ha creado correctamente”.

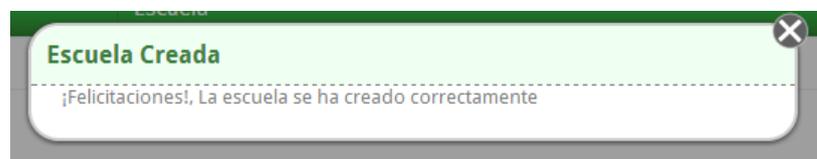


Ilustración 118: Página Web – mensaje: ingreso correcta de la escuela

- El sistema visualiza todas las escuelas creadas.



Seleccionar	Código	Escuela	Parroquia
<input type="radio"/>	01H00074	AURELIO AGUILAR VÁZQUEZ	CUENCA
<input type="radio"/>	01H01181	MANUEL JESUS AYABACA	EL CABO
<input type="radio"/>	14H00596	VICTOR HUGO CASTILLO VELIN	MACAS
<input type="radio"/>	01H00759	ABDON CALDERON	RICOURTE
<input type="radio"/>	01H01061	MIGUEL ANGEL GALARZA	SAN JUAN

Ilustración 119: Página Web – lista de todas las escuelas ingresadas

Anular

- Si alguna escuela fue creada de manera errónea, puede ser anulada, el usuario selecciona la escuela que va a eliminar y después presiona el botón Anular.



Seleccionar	Código	Escuela	Parroquia
<input type="radio"/>	01H00074	AURELIO AGUILAR VÁZQUEZ	CUENCA
<input type="radio"/>	01H01181	MANUEL JESUS AYABACA	EL CABO
<input type="radio"/>	14H00596	VICTOR HUGO CASTILLO VELIN	MACAS
<input type="radio"/>	01H00759	ABDON CALDERON	RICOURTE
<input checked="" type="radio"/>	01H01061	MIGUEL ANGEL GALARZA	SAN JUAN

Ilustración 120: Página Web – selección de la escuela a anular

- El sistema lanza el siguiente mensaje, si estamos seguros de anular presionamos el botón Aceptar, caso contrario presionamos Cancelar.

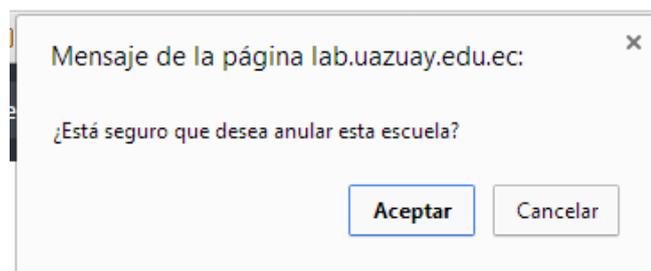


Ilustración 121: Página Web – mensaje de confirmación

- Si la escuela se anuló correctamente el sistema visualiza el siguiente mensaje: “La escuela se ha anulado correctamente”.

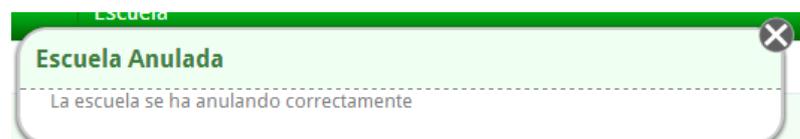


Ilustración 122: Página Web – mensaje: anulación correcta de la escuela

3.5.6.5. Subir margaritas

- Cuando el usuario presiona la opción subir margaritas, el sistema visualiza una pantalla donde se deben ingresar tres imágenes de margaritas, de las tres imágenes una es diferente (correcta), el usuario debe seleccionar la opción Correcta, además se debe seleccionar la expresión.

Subir margaritas
 Seleccione las imágenes que desea subir (3 margaritas), indique cuál es la correcta

Seleccione las imágenes que desea subir (3 margaritas), indique cuál es la correcta

EXPRESIÓN:

Ningún archivo seleccionado

IMAGEN 1
 Vista Previa: [Seleccione una imagen]

CORRECTO:

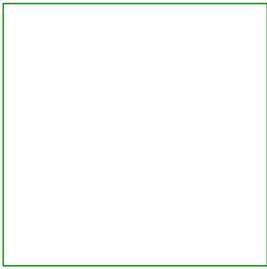


Ilustración 123: Página Web – pantalla de selección de imágenes

Subir margaritas
 Seleccione las imágenes que desea subir (3 margaritas), indique cuál es la correcta

Seleccione las imágenes que desea subir (3 margaritas), indique cuál es la correcta

EXPRESIÓN:

DD1.jpg

IMAGEN 1
 Vista Previa: DD1.jpg (Tipo: image/jpeg)
 Tamaño: 30367 bytes

CORRECTO:



Ilustración 124: Página Web – selección de la expresión e imagen a ingresar

- Cuando todas las margaritas se visualicen, el usuario almacena la información presionando el botón Subir imágenes.

Seleccionar archivo DD2.jpg

IMAGEN 2
Vista Previa: DD2.jpg (Tipo: image/jpeg)
Tamaño: 30515 bytes

CORRECTO:



Seleccionar archivo DD3.jpg

IMAGEN 3
Vista Previa: DD3.jpg (Tipo: image/jpeg)
Tamaño: 30446 bytes; Medidas: 200x200

CORRECTO:



Subir imágenes

Ilustración 125: Página Web – almacenamiento de las imágenes subidas

Manuales de Usuario – Android e iOS

La aplicación del Test de Margarita puede ser descargada desde cualquier tablet Android superior a 7 pulgadas y desde cualquier tablet iOS.

3.5.7. Descargar aplicación Android

- Para descargar la aplicación desde un dispositivo Android, el usuario accede al Play Store de google.

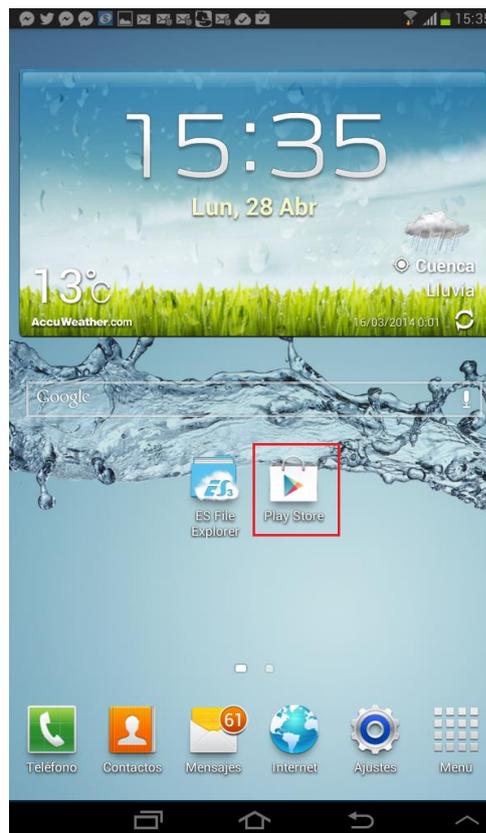


Ilustración 126: Android – play store – ícono de la tienda virtual de android

- El usuario debe digitar **Test de Margaritas** y presionar el botón buscar, se visualiza una pantalla donde encontramos el test junto con su logo, seleccionamos esta opción para leer una descripción y valorar el test.

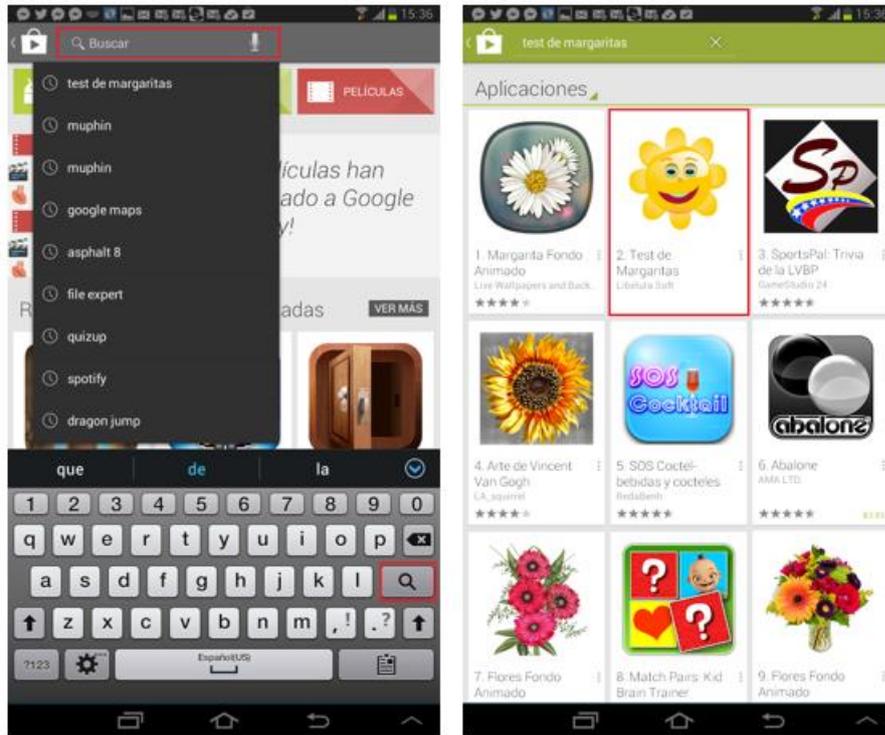


Ilustración 127: Android – play store – búsqueda de la aplicación Test de Margaritas

- El usuario presiona el botón INSTALAR, para que se visualice una pantalla con permisos.

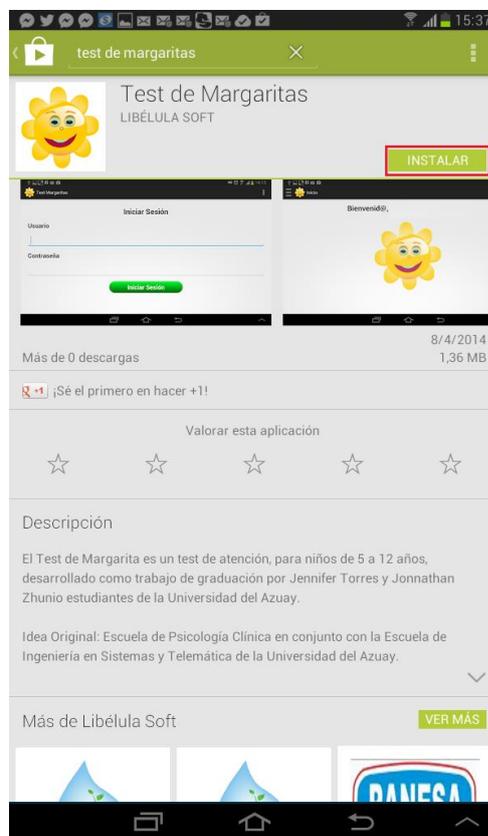


Ilustración 128: Android – play store – selección del botón de instalación

- Al presionar el botón ACEPTAR, se permite que la aplicación instale todos los permisos necesarios para su funcionamiento. Si todos los pasos de instalación son correctos, se comienza a instalar la aplicación en nuestro dispositivo.

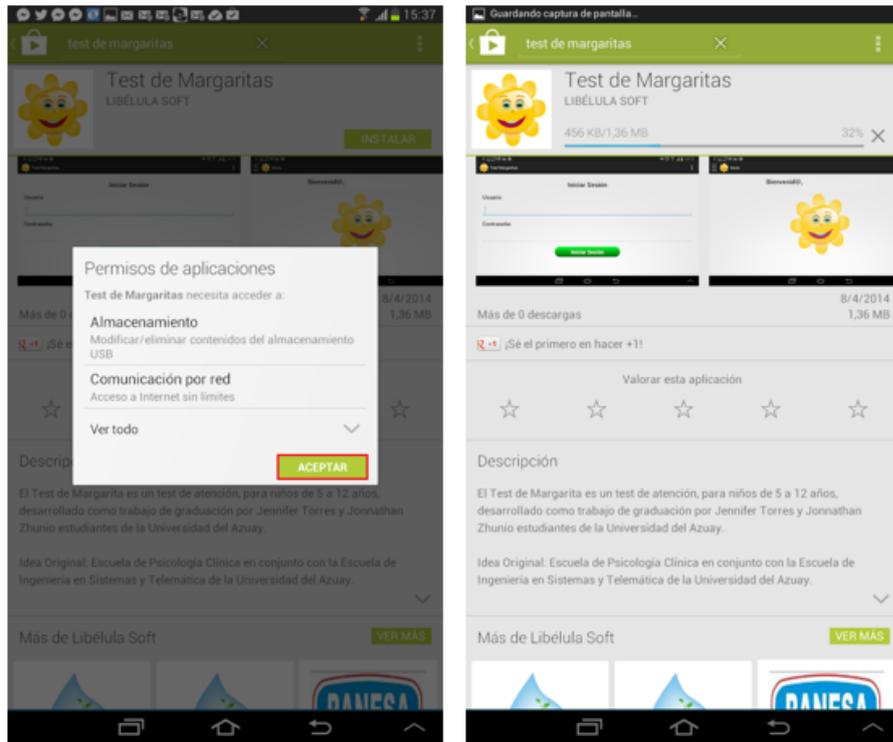


Ilustración 129: Android – play store – permisos e inicio de la instalación

- Si la aplicación se descargó correctamente, se muestra una pantalla donde podemos abrir la aplicación. Otra forma para ejecutar la aplicación, es buscándola en nuestro menú y seleccionando el test de margaritas.

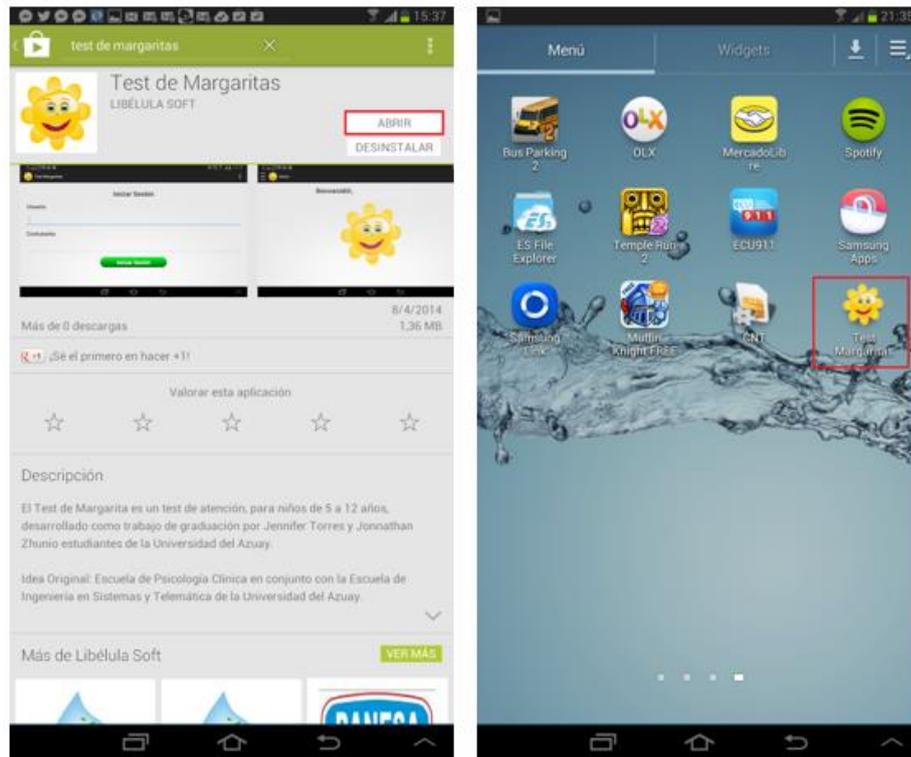


Ilustración 130: Android – play store – instalación completa, ícono de la aplicación Test de Margaritas

3.5.8. Descarga en iOS

La aplicación en iOS se encuentra en fase Beta, por ende, no está disponible para su descarga. De ser requerida, esta aplicación deberá ser probada en una tablet por sujetos de evaluación reales, para que en un futuro se pueda lanzar una versión estable.

3.5.9. Iniciar Sesión

- Para comenzar a utilizar el Test de Margaritas, el usuario debe ingresar nombre de usuario y contraseña, luego presiona el botón Iniciar Sesión. (usuario: prueba, contraseña: prueba)

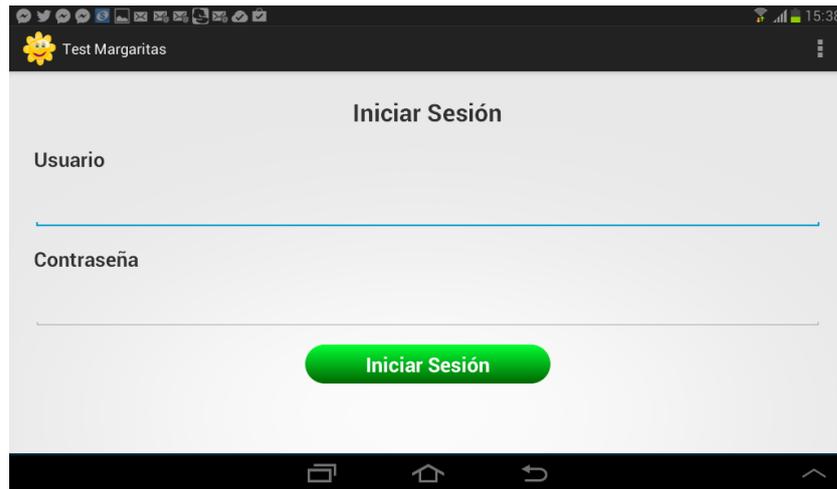


Ilustración 131: Android – inicio de sesión

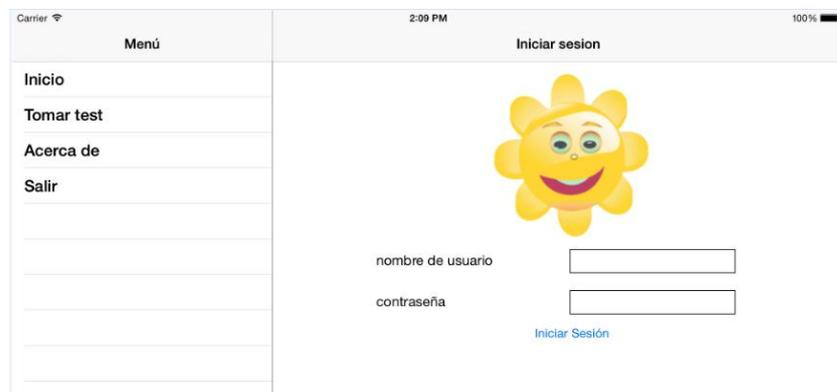


Ilustración 132: iOS - inicio sesión

- En Android la primera vez que inicia sesión el usuario debe tener conexión a internet, caso contrario el sistema visualiza el siguiente mensaje de error: “La primera vez que inicia sesión, debe tener conexión a internet”.
- En iOS, la aplicación siempre debe tener conexión a internet.

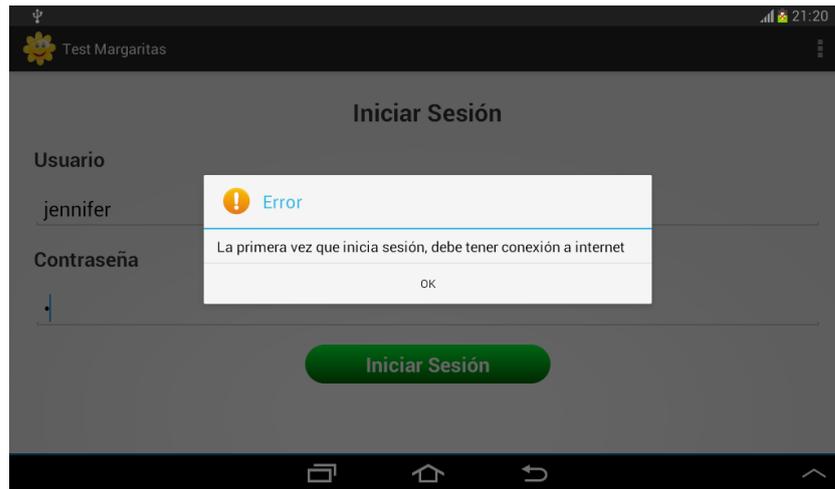


Ilustración 133: Android - mensaje: La primera vez que inicia sesión, debe tener conexión a internet

- Si el nombre de usuario o contraseña son incorrectos, el sistema visualiza el siguiente mensaje de error: “Usuario/Contraseña son incorrectos”.

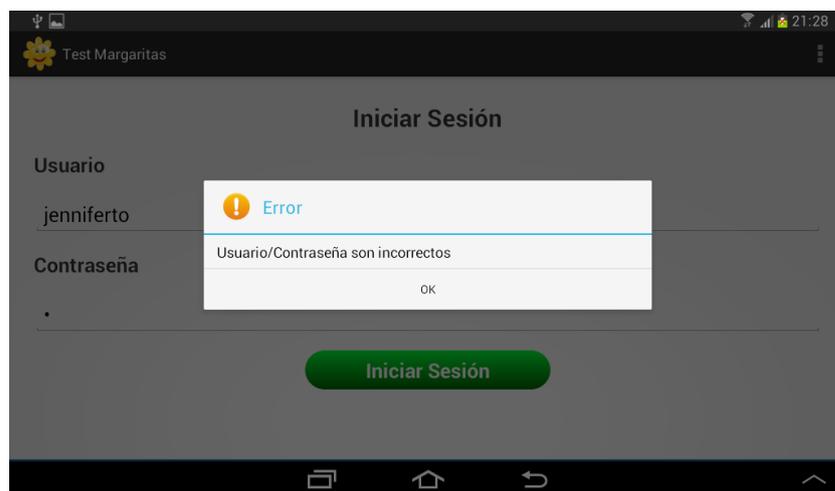


Ilustración 134: Android - mensaje: Usuario/Contraseña son incorrectos

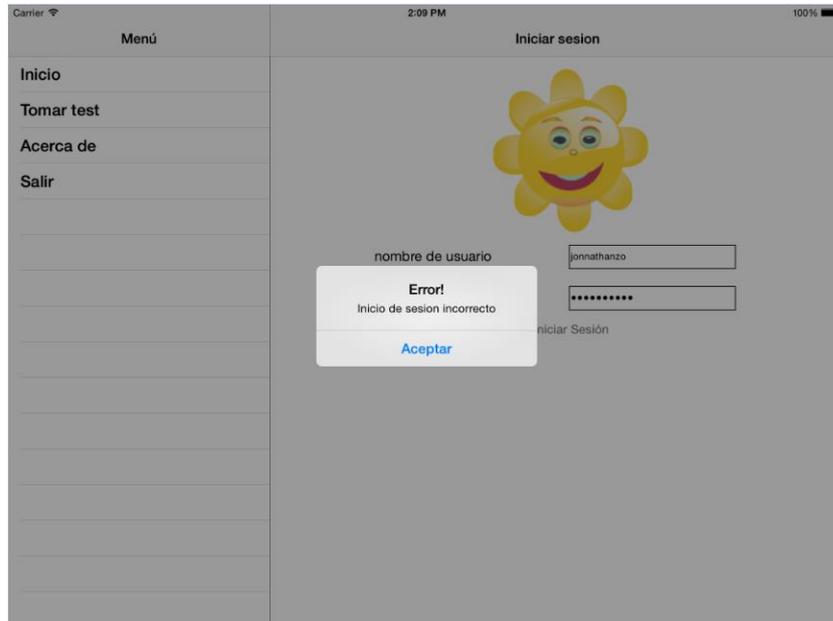


Ilustración 135: iOS - mensaje: Inicio de sesión incorrecto

- En Android si los datos ingresados son correctos, la aplicación comienza a copiar las tablas de MySQL a SQLite (propia del dispositivo), dependiendo de los datos ya ingresados en MySQL se puede tardar varios minutos.

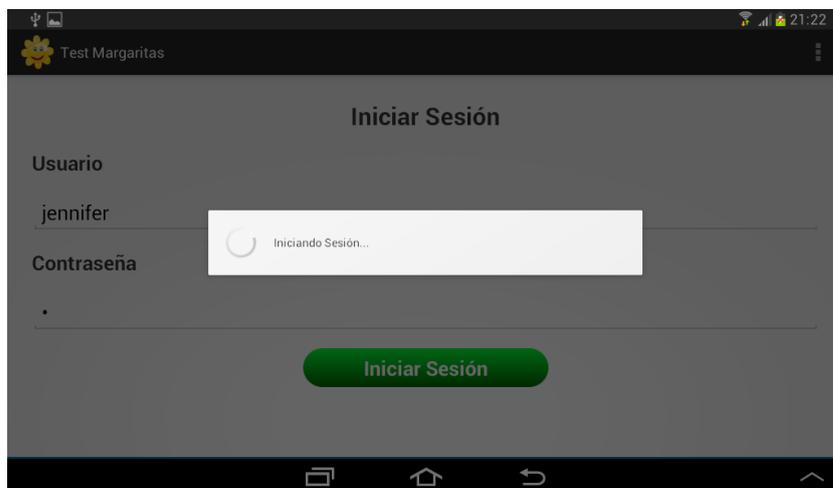


Ilustración 136: Android - iniciando sesión

- En iOS si los datos ingresados son correctos, la aplicación utiliza Web Service para mostrar todas las fichas ingresadas.

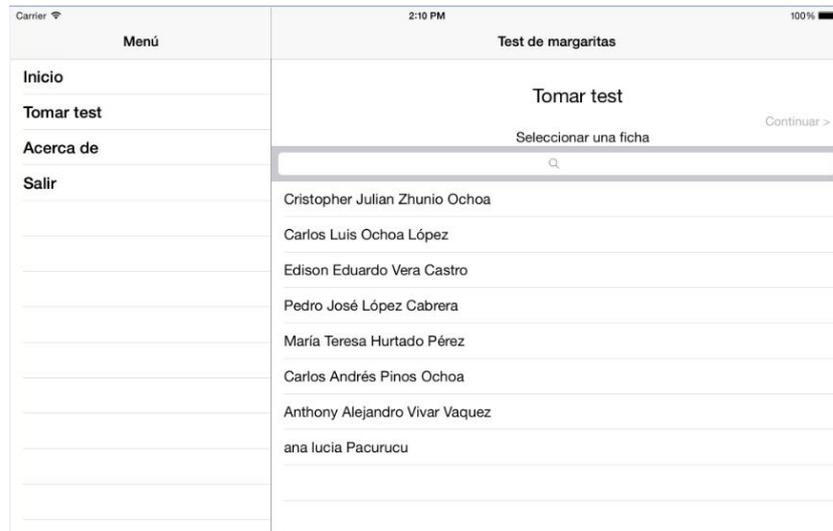


Ilustración 137: iOS - pantalla de inicio

3.5.10. Menú Acerca de

- Si el usuario presiona el siguiente botón:

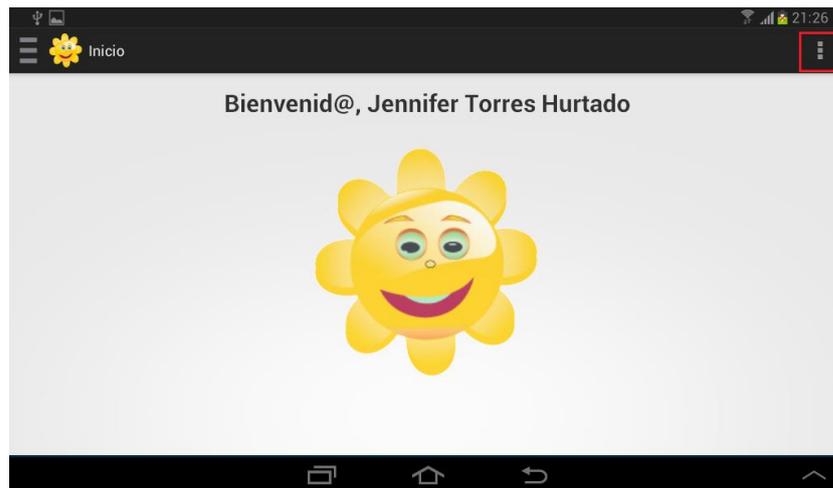


Ilustración 138: Android - pantalla de inicio

- El sistema presenta la opción Acerca de.

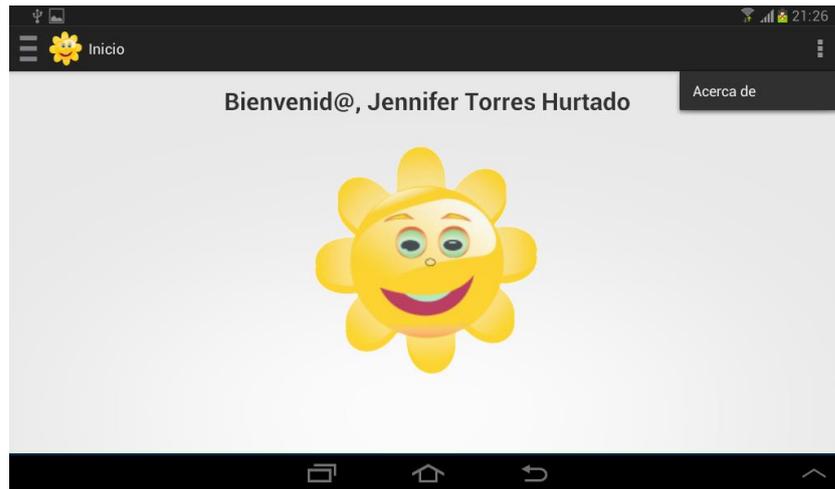


Ilustración 139: Android - submenú acerca de

- Si presionamos esta opción, el sistema visualiza la información referente al desarrollo del test.

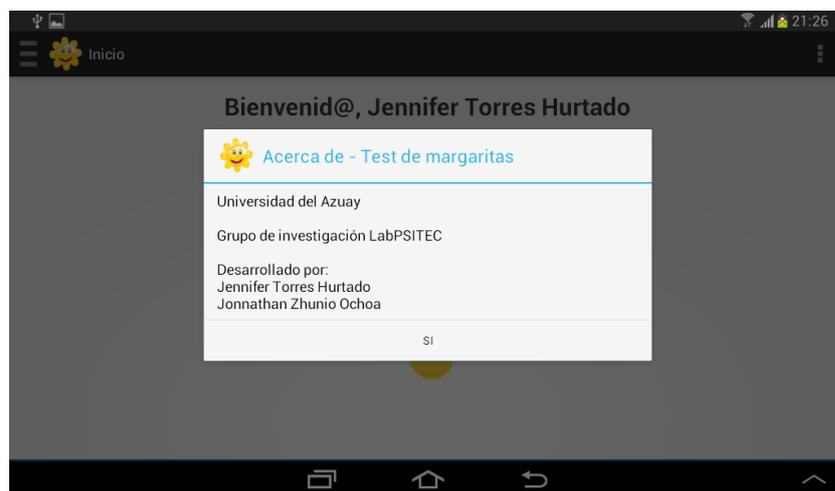


Ilustración 140: Android - pantalla acerca de

3.5.11. Menú Inicio

- Esta es la pantalla principal, el sistema da la bienvenida al usuario mostrando su nombre y apellidos.

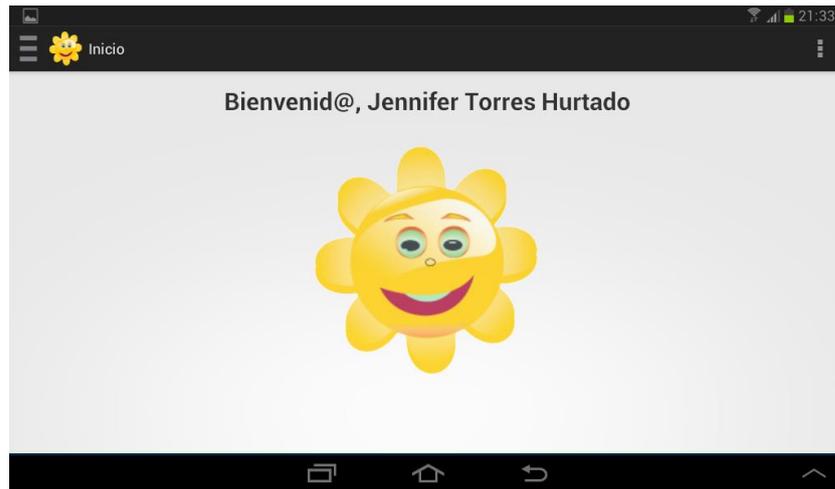


Ilustración 141: Android - pantalla de bienvenida

- Si el usuario presiona el botón:  el sistema visualiza un menú con las siguientes opciones: Inicio, Fichas, Nueva Ficha, Tomar Test, Sincronizar y Salir.

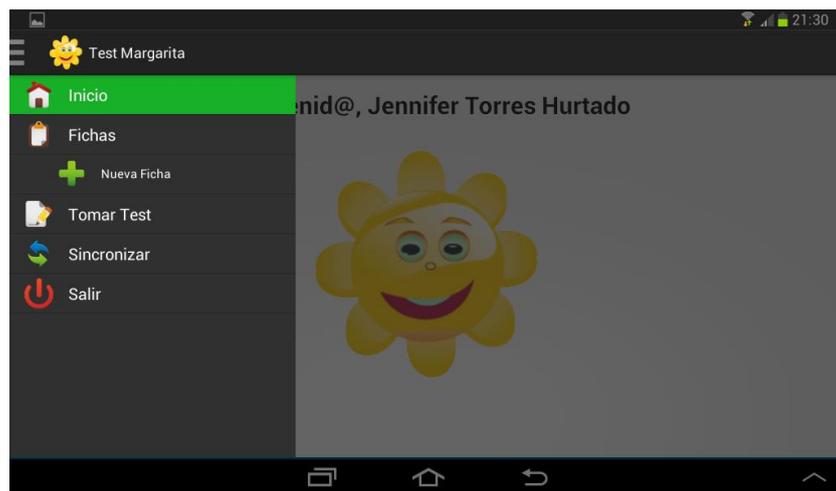


Ilustración 142: Android - menú principal

3.5.12. Menú Fichas

- La primera vez que se ejecuta este menú, el sistema visualiza un mensaje en el que se indica que no existe fichas creadas.

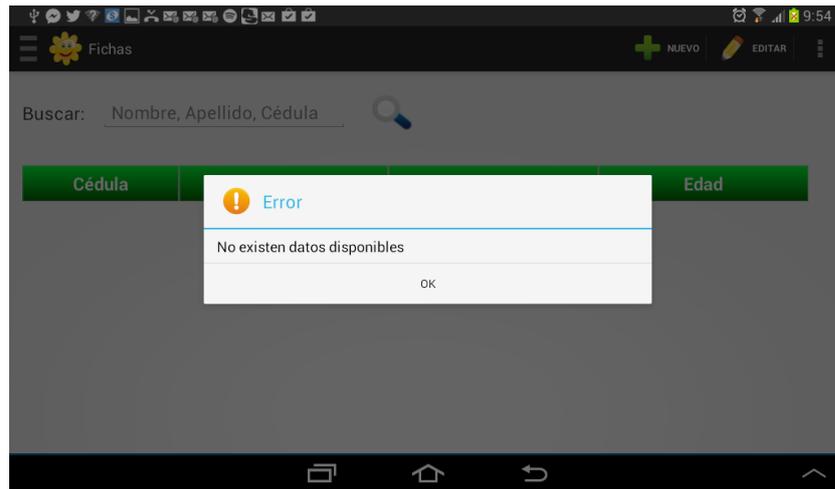


Ilustración 143: Android – mensaje: No existen datos disponibles

- Si el usuario ha ingresado previamente información, se visualiza una pantalla donde puede encontrar todas las fichas creadas, además las siguientes opciones: buscar, nuevo y editar.



Ilustración 144: Android - lista de todas las fichas ingresadas

Buscar

- Se puede realizar la búsqueda por Nombre, Apellido y Cédula. Se digita un carácter como mínimo y se presiona la imagen buscar, el sistema visualiza una pantalla con las fichas que tienen incluido el carácter digitado sin importar donde esté ubicado.

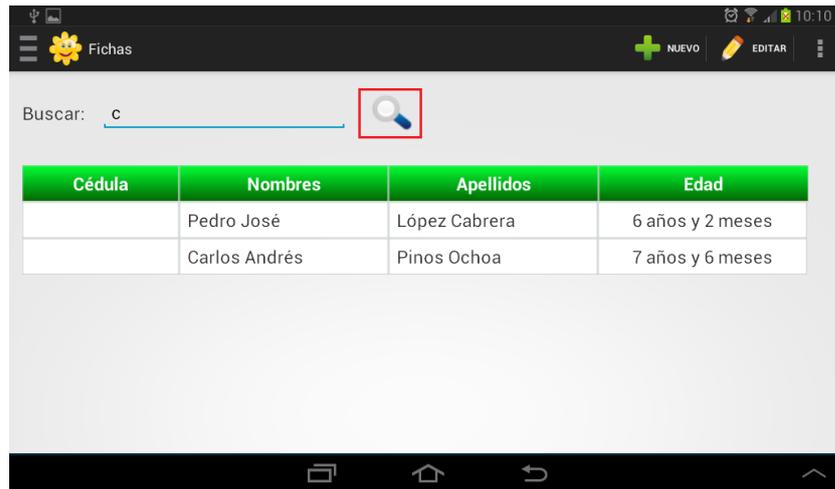


Ilustración 145: Android - ingreso de caracteres a buscar

Editar

- Se selecciona la ficha y se presiona el botón Editar.



Ilustración 146: Android - selección de la ficha a editar

- A continuación se visualiza una pantalla con toda la información de la ficha seleccionada, se modifican los campos necesarios y se presiona el botón guardar. Si la información es almacenada de forma correcta, el sistema visualiza el siguiente mensaje: “Datos Guardados Correctamente”.



Ilustración 147: Android - modificación de datos de la ficha

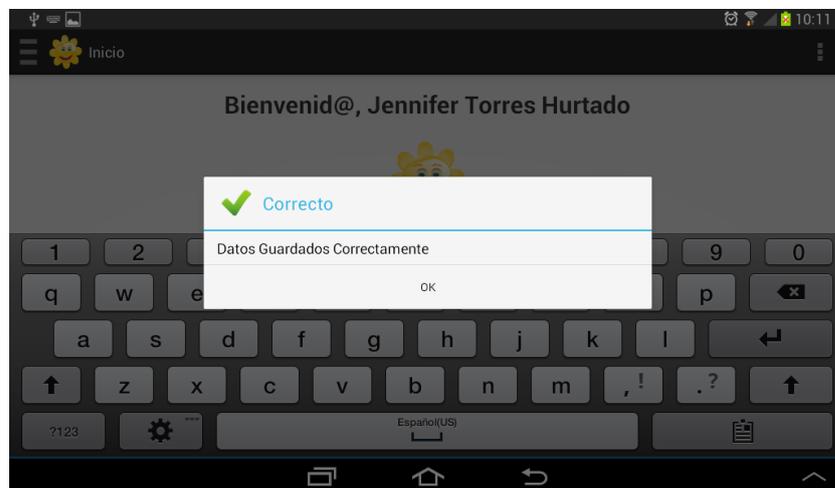


Ilustración 148: Android - mensaje: datos guardados correctamente

Nuevo

- El usuario puede crear una nueva ficha de dos maneras: seleccionando el botón nuevo o buscando en el menú principal la opción Nueva Ficha.

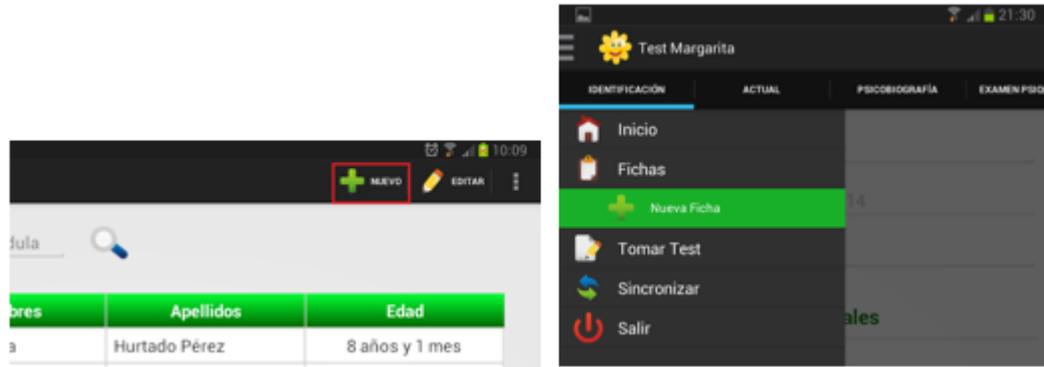


Ilustración 149: Android - ingresar nueva ficha

Menú Nueva Ficha

- En este menú el sistema presenta cuatro pestañas: Identificación, Actual, Psicobiografía y Examen Psiquiátrico. El usuario debe llenar todos los campos necesarios (*) -se encuentran únicamente en la pestaña Identificación-, a continuación presiona el botón Guardar para se cree de forma correcta la ficha.

Datos Generales

Fecha nacimiento:* 06/03/2005 Elegir Fecha

Edad 6 años y 2 meses

Número de hermanos: 2

Datos Escolares

Provincia:* AZUAY

Cantón:* CUENCA

Parroquia:* CUENCA

Escuela:* AURELIO AGUILAR VÁZQUEZ

Nivel Escolar:* 4to de Básica

Padre

Nombre: _____

Nacionalidad: _____

Ilustración 150: Android - ingreso de una nueva ficha

- Si la información se almacenó de forma correcta el sistema visualiza el siguiente mensaje: “Datos Guardados Correctamente”.

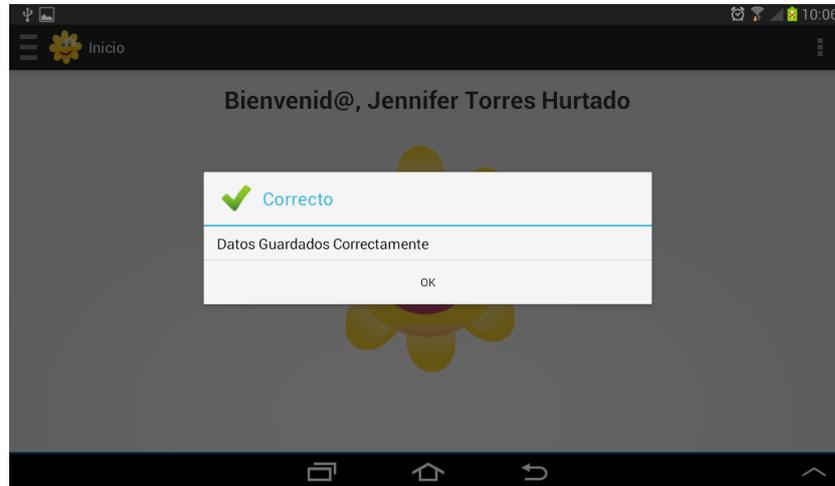


Ilustración 151: Android - mensaje: datos guardados correctamente

- Si se desea salir de la ficha sin almacenar la información se presiona el botón salir, el sistema muestra un mensaje de confirmación, si está seguro de salir se presiona el botón SI, caso contrario se presiona NO.



Ilustración 152: Android - mensaje confirmación

3.5.13. Menú Tomar Test Android

- Cuando presionamos el menú Tomar Test, el sistema visualiza todos los test creados previamente por un usuario administrador. Seleccionamos el test que va a desarrollar el sujeto evaluación.



Ilustración 153: Android - selección del test

- El sistema muestra todas las fichas creadas, el usuario debe seleccionar al sujeto que va a realizar el test y presionar el botón siguiente.

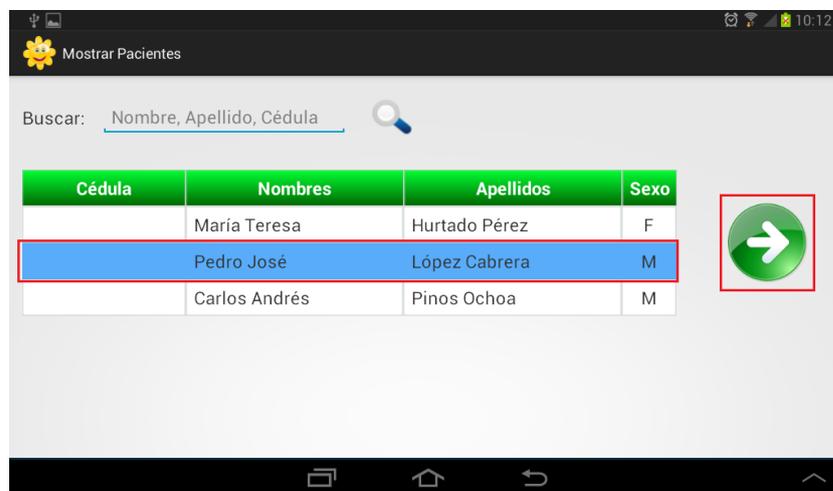


Ilustración 154: Android - selección del paciente

- Si todos los pasos son correctos, el sistema visualiza las instrucciones y a continuación el test.



Ilustración 155: Android - instrucciones y toma del test

- Cuando el sujeto evaluación termina el test, el sistema muestra el siguiente mensaje: “Felicitaciones, has terminado el Test”.



Ilustración 156: Android - mensaje confirmación

3.5.14. Menú Tomar Test en iOS

- Cuando presionamos el menú Tomar Test, el sistema visualiza todas las fichas existentes.

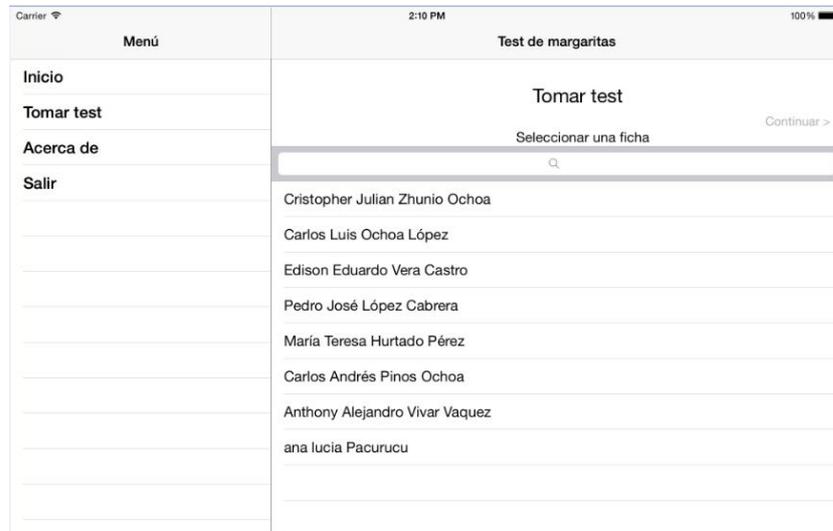


Ilustración 157: iOS - lista de todos los pacientes

- Se puede realizar búsquedas por fichas, se ingresa dos caracteres como mínimo y el sistema busca todas las fichas que contengan a esos caracteres.

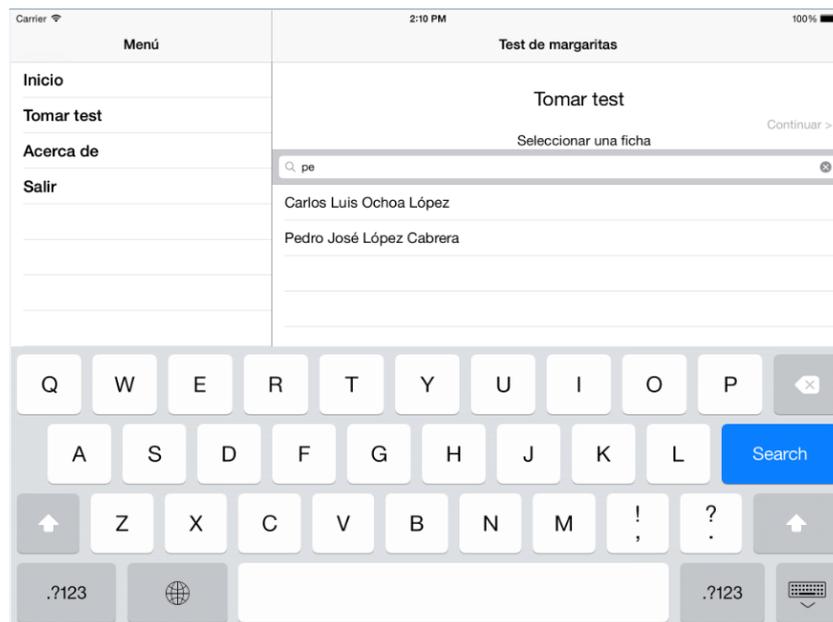


Ilustración 158: iOS - buscar pacientes

- Seleccionamos la ficha deseada y presiona el botón Continuar.

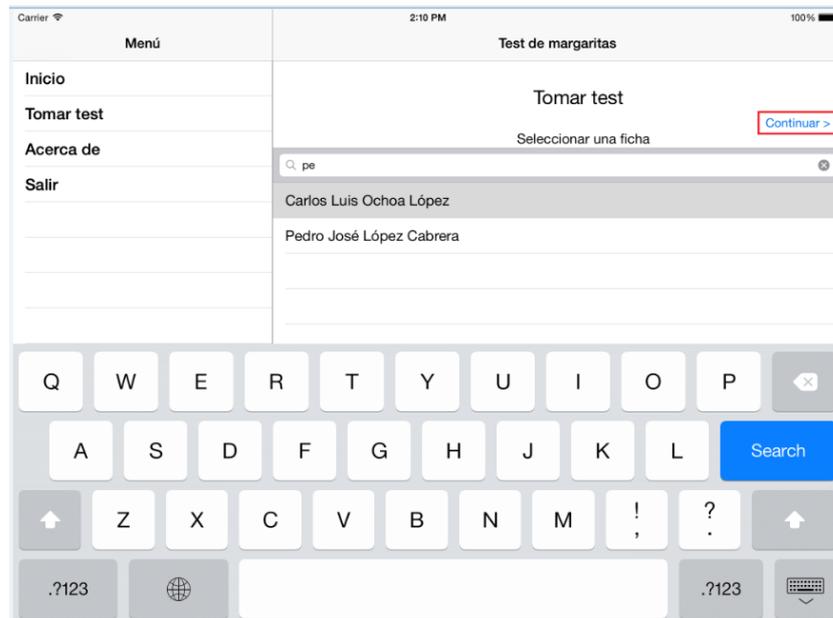


Ilustración 159: iOS - botón continuar

- El sistema muestra todos los test creados, el usuario debe seleccionar el test y presionar el botón siguiente.

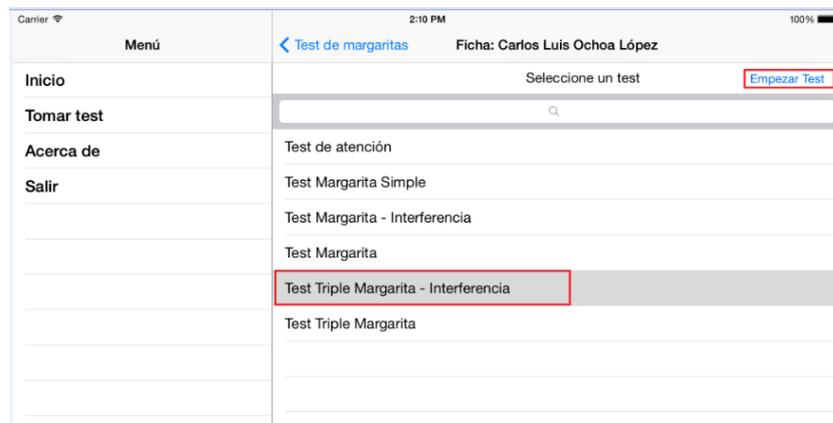


Ilustración 160: iOS - seleccionar el test

- Si todos los pasos son correctos, el sistema muestra el test.

Pregunta 2

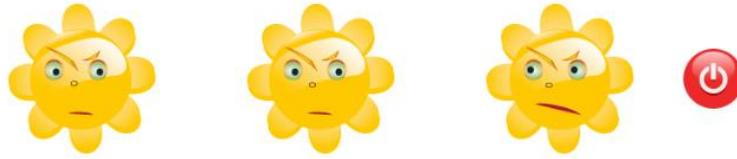


Ilustración 161: iOS - mostrar el test

- Cuando el sujeto evaluación termina el test, el sistema muestra el siguiente mensaje: “Test Terminado, se han guardado los datos”.

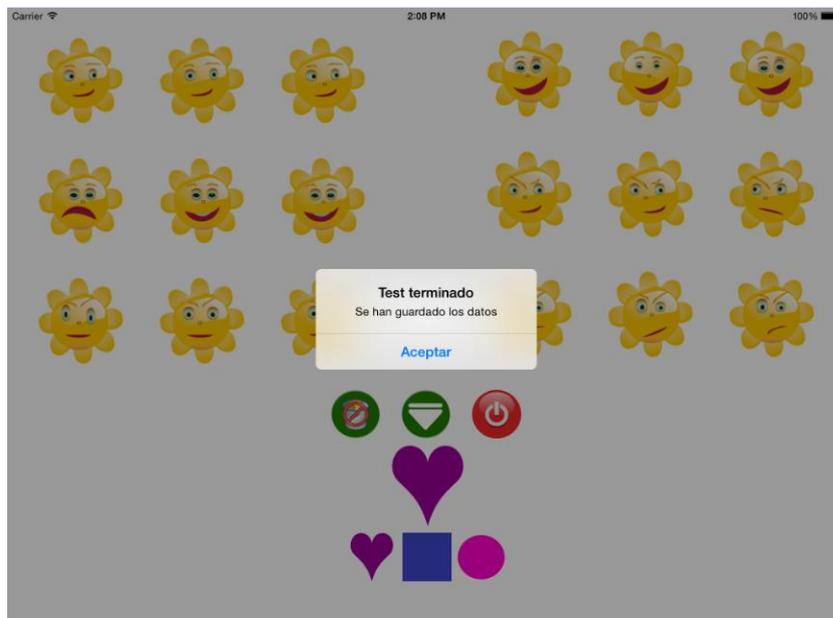


Ilustración 162: iOS - mensaje de confirmación

3.5.15. Menú Sincronizar

- En el menú sincronizar el sistema presenta cuatro opciones: Actualizar, Subir Información, Descargar Fichas, Vaciar Fichas. Para ejecutar cualquiera de estas opciones, el dispositivo debe tener conexión a internet.



Ilustración 163: Android - menú sincronizar

- Actualizar.- cuando se haya ingresado o modificado datos principales como test, interferencias y usuarios en la página web, se puede sincronizar la información de la página web al dispositivo móvil con la ayuda de esta opción. Si la información se ha actualizada de manera correcta el sistema visualiza el siguiente mensaje: “Datos Actualizados Correctamente”.

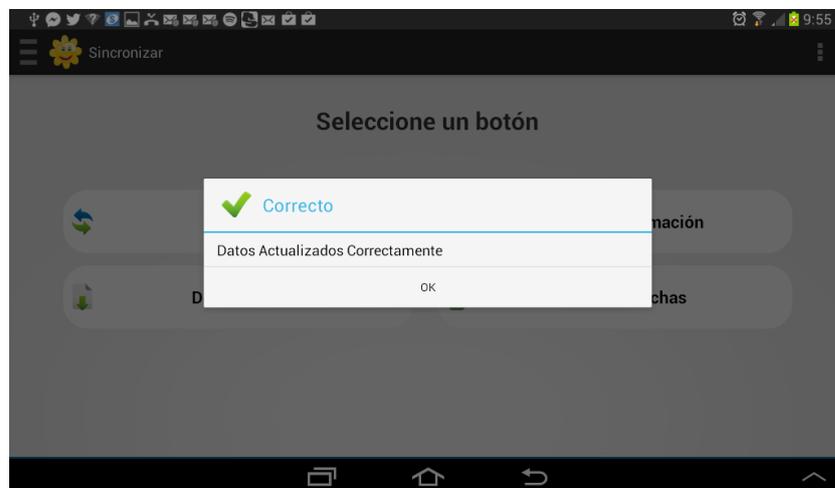


Ilustración 164: Android - mensaje confirmación datos actualizados

- Subir Información.- cuando el usuario ingrese fichas y el sujeto evaluación realice test, se almacenan en la base de datos de SQLite. Todos los campos modificados o ingresados deben ser subidos a la base de datos principal (MySQL), puesto que, los datos almacenados en SQLite van a ser eliminados por el mismo usuario. Ocupamos la opción Subir Información, para copiar los datos de SQLite a MySQL

con la ayuda de un Web Service (en PHP). Si la información se subió de forma correcta el sistema visualiza el siguiente mensaje: “Información Subida Correctamente”.



Ilustración 165: Android - mensaje confirmación información subida

- Descargar Fichas.- cuando el usuario necesite descargar varias fichas almacenadas en la base de datos de MySQL, puede realizar esta acción, filtrando la información por datos de escuela: Provincia, Cantón, Parroquia y Escuela. Si las fichas se descargan correctamente el sistema visualiza el siguiente mensaje: “Datos Descargados Correctamente”.



Ilustración 166: Android - descargar fichas por escuela



Ilustración 167: Android - mensaje confirmación descarga correcta

- Vaciar Ficha.- cuando el usuario ya no necesita evaluar a un sujeto puede eliminar su ficha, dado que la información de este sujeto ya está almacenada en la base de datos principal (MySQL). Si las fichas se eliminaron de forma correcta, el sistema visualiza el siguiente mensaje: “Fichas Vacías Correctamente”.



Ilustración 168: Android - mensaje confirmación vaciar fichas

CAPÍTULO 4: PRUEBAS

Para garantizar el correcto funcionamiento de la página web y aplicación Android se realizaron diferentes pruebas de funcionamiento frente a posibles situaciones, al funcionamiento interno de la base de datos de ambos sistemas.

En las pruebas realizadas en la interfaz se verificó que las respuestas sean las esperadas, que se validen los datos correctamente, esto permite que la integridad de la información se mantenga.

En las pruebas realizadas a la base de datos se verificó que los datos se almacenen de forma correcta, el buen funcionamiento de los triggers y que las bases de datos de SQLite y MySQL se sincronicen.

4.1. Página Web

Seguridad:

- Se solicita en la pantalla de inicio el ingreso de un nombre de usuario y una contraseña, esto permite que únicamente usuarios registrados puedan tener acceso a la página.
- Se utiliza captcha que ayuda a determinar si un usuario es o no humano.
- Se verifica que el usuario tenga los suficientes permisos para realizar una acción.

Compatibilidad:

- La aplicación web fue probada en los siguientes navegadores:
 - Google Chrome versión 35.0.1916.114 m
 - Safari versión 7.0.3
 - Mozilla Firefox versión 29.0

Componente	Navegadores			
	Mozilla Firefox	Google Chrome	Apple Safari	Internet Explorer
Iniciar sesión	√	√	√	√
Mi perfil	√	√	√	√
Agendar citas	√	√	√	X

Fichas	√	√	√	X
Resultados test	√	√	√	√
Tomar test	√	√	√	√
Motivo consulta	√	√	√	X
Administrar test	√	√	√	X
Administrar usuarios	√	√	√	X
Administrar escuelas	√	√	√	X
Subir margaritas	√	√	√	√

Tabla 13: Tabla funcionamiento navegadores

4.2. Aplicación Android

Seguridad:

- Se solicita en la pantalla de inicio el ingreso de un nombre de usuario y una contraseña, esto permite que únicamente usuarios registrados puedan tener acceso a la página.
- Para que la información no se pierda cuando se vacíen las fichas, de forma transparente el sistema verifica que la información se encuentre actualizada en la base de datos MySQL del servidor, si no es el caso manda a almacenar la información.
- La aplicación está disponible únicamente para tablets de 7 pulgadas con versiones superiores a Android 3.0 Honeycomb (API Level 11).

Interacción:

- Cuando se sincroniza la información de la base de datos SQLite a MySQL, puede ocurrir un retardo por el tiempo de acceso a la red.
- El tiempo de retardo depende del ancho de bando, pues se utilizan conexión a internet para la sincronización.

Compatibilidad:

- La aplicación Android fue probada en las siguientes tablets de 7 pulgadas:
 - Samsung Galaxy Tab 3 versión de Android 4.1.2 con una resolución de pantalla de 1.024 x 600 WSVGA.

Componente	Samsung Galaxy Tab 3
Descargar aplicación	√
Iniciar sesión	√
Acerca de	√
Inicio	√
Fichas	√
Nueva ficha	√
Tomar test	√
Sincronizar	√

Tabla 14: Tabla funcionamiento aplicación Android

4.3. Aplicación iOS

La aplicación iOS, para ser sometida a pruebas en una tablet, necesita de una cuenta de desarrollador, porque es a través de esta que se puede publicar la aplicación y realizar pruebas en la misma, todo este proceso tiene un costo anual.

Dentro de esta aplicación podemos manifestar que se nos hizo imposible realizar las pruebas, puesto que no se justificaba la inversión económica en esta, debido a que el cliente tiene previsto -inicialmente- utilizar este trabajo, únicamente en la aplicación Android. Sin embargo, cabe explicar que este proyecto, se encuentra en una versión Beta, dicho de otra forma, está terminada pero no está probada.

Consideramos que, más adelante, si el cliente considera necesario, podrá realizar las pruebas de rigor.

CONCLUSIONES

Al culminar este trabajo se recalca que todos los objetivos planteados fueron cumplidos, en base a investigación y elaboración de software, se logró diseñar y desarrollar el Test de Margaritas con la utilización de diferentes tecnologías como Web, Android e iOS, estos contienen un sistema de gestión administrativa y un sistema de aplicación del test.

Las aplicaciones móviles utilizan dispositivos táctiles que se encargan de facilitar la toma del test en niños, pues presenta ventajas de facilidad de utilización frente a computadores. Además la aplicación Android está diseñada para funcionar con o sin conexión a internet, permite borrar información que no se va utilizar a futuro y contiene algunos aspectos de la gestión administrativa.

La aplicación web se encarga de la gestión administrativa, presenta ventajas como el acceso desde cualquier lugar, la gestión de usuarios y mantenimientos, y muestra los resultados obtenidos en la toma del test.

Este trabajo de investigación personifica la colaboración multidisciplinaria entre dos ramas (la tecnología y la psicología), además representa un paso importante de la investigación en nuestro medio, puesto que se proporcionan herramientas que a futuro arrojarán resultados que van a ser utilizados en nuevos estudios con la finalidad de generar una línea de investigación fuerte en la Universidad del Azuay.

RECOMENDACIONES

Para la aplicación Android se recomienda que se la descargue directamente desde Play Store de Google, y sea ejecutada en tablets de 7 pulgadas con una resolución de pantalla de 1.024 x 600 WSVGA. También se debe tener claro que la primera vez que el usuario inicia sesión en la aplicación, es obligatorio que tenga conexión a internet pues el programa necesita descargar los datos.

Para la aplicación iOS se debe tener presente que no estará publicada, pues el costo de publicación es alto y por el momento, los test serán aplicador por parte de la Universidad del Azuay en dispositivos Android.

Para la aplicación web se recomienda efectuar un respaldo de la base de datos y que se disponga de un subdominio dentro del servidor de la Universidad del Azuay. También se recomienda que no se acceda a la página desde el navegador Internet Explorer.

GLOSARIO

TPD-C.- Test de Percepción de Diferencias – Caras – es un test que evalúa las aptitudes perceptivo-imaginativas, se apoyan en principios de semejanza-diferencias y utilizan material impreso. (Thurstone & Yela, Caras-R Test de Percepción de Diferencias - Revisado., 2012)

Rapidez de percepción.- La habilidad para identificar elementos semejantes, así como para discriminar elementos diferentes. (Fernández Muñoz, 2011)

Inteligencia espacial.- Es la capacidad de pensar en tres dimensiones. Permite percibir imágenes externas e internas, recrearlas transformarlas o modificarlas. (Fernández Muñoz, 2011)

ICI.- Índice de Control de Impulsividad, este índice expresa el nivel de control de impulsividad del sujeto a la hora de ejecutar una tarea, se trata del porcentaje de aciertos correctos que obtiene una persona en los ítems que responde. (Thurstone L. L., 2013)

Psicobiografía.- Se refiere a la exploración psicológica de la vida de una persona. (Aiken, 2003)

Base de datos.- Es un sistema computarizado para guardar registros, es decir, la finalidad general es almacenar información y pedir a los usuarios recuperar y actualizar esta información con base en peticiones. (Date & Ruiz Faudón, 2001)

SQL.- Structured Query Language (en español Lenguaje de Consulta Estructura), es un lenguaje utilizado en la mayor parte de sistemas de gestión de bases de datos de carácter relacional. (Quintana, Marqués, Aliaga, & Arambur, 2008)

SGBDR.- Sistema de Gestión de Base de Datos Relacional, es un conjunto de programas que permiten el almacenamiento, modificación y extracción de información contenida en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. (Quintana, Marqués, Aliaga, & Arambur, 2008)

ISAM.- Son siglas de Indexed Sequential Access Method (Método de Acceso Secuencial Indexado), se trata de un método para almacenar información a la que se pueda acceder rápidamente. (Hinz, y otros, MySQL Documentation, 2014)

MyISAM.- Es el motor de almacenamiento por defecto. Se basa en el código ISAM pero tiene muchas extensiones útiles. (Hinz, y otros, MySQL Documentation, 2014)

Web Service.- Es una tecnología que utiliza un conjunto de protocolos y estándares que encapsulan funcionalidades y las publican para que puedan ser accedidas por otras aplicaciones, mediante una serie de protocolos basados en XML. (Bravo Santos & Redondo Duque, 2005)

TRIGGER.- Un trigger o disparador es un objeto con nombre dentro de una base de datos el cual se asocia con una tabla y se activa cuando ocurre en ésta un evento en particular. (Hinz, y otros, MySQL Documentation, 2014)

SDK de Android.- Software Development Kit (Kit de Desarrollo de Software), proporciona las bibliotecas y las herramientas de desarrollo necesarias para crear, probar y depurar aplicaciones para Android. (Android Developers, 2014)

Json.- JavaScript Object Notation, es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. (Json Corporation, 2014)

Captcha.- Es un programa que protege a los sitios web de los robots, mediante la generación y la clasificación de pruebas que los humanos pueden pasar pero los programas informáticos actuales no pueden. Por ejemplo, los seres humanos pueden leer texto distorsionado, pero los programas informáticos actuales no pueden. (The Official CAPTCHA Site, 2014)

Web Apache.- El Proyecto Apache HTTP Server es un esfuerzo para desarrollar y mantener un servidor HTTP de código abierto para sistemas operativos modernos, incluyendo UNIX y Windows NT. El objetivo de este proyecto es proporcionar un

servidor seguro, eficiente y extensible que proporcione servicios HTTP en sincronización con los estándares HTTP actuales. (Apache Group, 2014)

Open Source.- Software de código abierto es un software que puede ser utilizado libremente, cambiado, y compartido (en forma modificada o no modificada) por cualquier persona. Es hecho por muchas personas, y se distribuye bajo licencias que cumplen con la definición de código abierto. (Open Source Community, 2014)

UML.- Unified Modeling Language (Lenguaje de Modelado Unificado) es la especificación más utilizada de OML (Object Management Group), es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. No sólo se utilizará para la estructura, comportamiento y arquitectura de una aplicación, sino también para procesos de negocio y la estructura de datos. (Object Management Group, 2014)

BIBLIOGRAFÍA

- Aiken, L. R. (2003). *Tests psicológicos y evaluación* (Undécima ed.). México: PEARSON Educación. Obtenido de <http://books.google.com.ec/books?id=2LvyL8JEDmQC&printsec=frontcover#v=onepage&q&f=false>
- Alejandro Ramirez, P. V. (2011). *ArgoUML: Manual de Usuario Tutorial y descripción de referencia*. Recuperado el 19 de 11 de 2013, de Tigris.org: <http://argouml-stats.tigris.org/daily-userdoc/es/pdf/manual/manual.pdf>
- Android Developers. (24 de Abril de 2014). *Android Developers*. Obtenido de <http://developer.android.com/tools/projects/index.html>
- Android Developers. (24 de Abril de 2014). *Android Developers*. Obtenido de <http://developer.android.com/training/multiscreen/screensizes.html>
- Android Developers. (28 de Mayo de 2014). *Android Developers*. Obtenido de <http://developer.android.com/sdk/index.html>
- Apache Group. (29 de Mayo de 2014). *Apache*. Obtenido de <http://httpd.apache.org/>
- AUBRY, C. (2012). *HTML5 y CSS3 - Revolucione el diseño de sus sitios web*. Barcelona: Ediciones ENI. Obtenido de <http://goo.gl/WvHPAG>
- Bravo Santos, C., & Redondo Duque, M. Á. (2005). *Sistemas interactivos y colaborativos en la web*. España: Univ de Castilla La Mancha. Obtenido de <http://goo.gl/R2LN9A>
- Brooks, D. R. (2011). *Guide to HTML, JavaScript and PHP: For Scientists and Engineers*. New York: Springer. Obtenido de <http://goo.gl/C0hHwG>
- Cancela García, L., & Ostos Lobo, S. (22 de Abril de 2014). *Software de Comunicaciones*. Obtenido de <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>
- Crespo, Eguílaz, R., Narbona, J., Perlta, F., & Reparaz, R. (2006). Medida de atención sostenida y del control de la impulsividad en niños: nueva modalidad de aplicación del Test de Percepción de Diferencias “Caras”.
- Cyril THIBAUD. (2006). *MySQL 5: Instalación, Implementación, Administración, Programación*. Barcelona: Ediciones Eni. Recuperado el 25 de Abril de 2014, de <http://goo.gl/h1hQCN>

- Damián, D. (2011). *HTML5: entienda el cambio, aproveche su potencial*. Buenos Aires: Fox Andina. Obtenido de <http://goo.gl/f5kLu1>
- Dark, V. J., & Johnston, W. A. (1982). In defense of intraperceptual theories of attention. *Journal Experimental Psychology: Human Perception and Performance*.
- Date, C. J., & Ruiz Faudón, S. L. (2001). *Introducción a los sistemas de bases de datos* (Séptima ed.). México: PEARSON Educación. Obtenido de <http://goo.gl/K1vYMw>
- ERIKA CAMACHO, o. (Abril de 2004). *ARQUITECTURAS DE SOFTWARE*. Recuperado el 4 de 12 de 2013, de <http://prof.usb.ve/lmendoza/Documentos/PS-6116/Guia%20Arquitectura%20v.2.pdf>
- Fernández Muñoz, Á. (2011). *Pruebas psicotécnicas* (3era ed.). Centro de Estudios Financieros.
- Fernández Pérez, G. (2013). *iOS, Todo lo que siempre has querido saber sobre tu iPhone y iPad*. Recuperado el 25 de Abril de 2014, de <http://goo.gl/SGoMbH>
- García, J. (1997). *Psicología de la atención*. Madrid: Síntesis.
- Gardner, H. (1983). *Multiple Intelligences*, Castellano: “Inteligencias múltiples”. Paidós.
- Gironés, J. T. (2013). *El gran libro de Android*. Barcelona: MARCOMBO S.A. Recuperado el 24 de Abril de 2014, de <http://goo.gl/NJlcUz>
- Hinz, S., DuBois, P., Stephens, J., Olson, P., Price, D., So, D., & Gilmore, E. (28 de Mayo de 2014). *MySQL*. Obtenido de <http://dev.mysql.com/doc/refman/5.0/es/myisam-storage-engine.html>
- Hinz, S., DuBois, P., Stephens, J., Olson, P., Price, D., So, D., & Gilmore, E. (25 de Abril de 2014). *MySQL Documentation*. Obtenido de <https://dev.mysql.com/doc/refman/5.0/es/myisam-storage-engine.html>
- iOS Developer. (25 de Abril de 2014). *iOS Developer*. Obtenido de <http://goo.gl/qviXdc>
- iOS Developer. (25 de Abril de 2014). *iOS Developer*. Obtenido de <http://goo.gl/mcMRMV>
- iOS Developer. (25 de Abril de 2014). *iOS Developer*. Obtenido de <http://goo.gl/eqxugF>

- iOS Developer. (25 de Abril de 2014). *iOS Developer*. Obtenido de <http://goo.gl/Gltbdi>
- iOS Developer. (26 de Abril de 2014). *iOS Developer*. Obtenido de <http://goo.gl/OIfCTR>
- James, W. (1980). En C. Lafebre, *DISEÑO DE UN MODELO DE ORIENTACIÓN PEDAGÓGICA A LOS DOCENTES PARA MEJORAR EL NIVEL DE ATENCIÓN EN NIÑOS/AS DE 5 A 10 AÑOS CON SÍNDROME DE DOWN* (pág. 52). Quito. Obtenido de http://repositorio.ute.edu.ec/bitstream/123456789/16883/1/53894_1.pdf
- Johnston, W., & Dark, V. (1986). Selective attention. En *Annual Review of Psychology* (págs. 43-75).
- Json Cooperation. (28 de Mayo de 2014). *Introducing JSON*. Obtenido de <http://www.json.org/>
- Kahneman, D. (1973). *Attention and effort*. Englewoods Cliffs, New Jersey: Prentice-Hall.
- Niekamp, R. (s.f.). *Software Component Architecture*. Obtenido de Institute for Scientific Computing: <http://congress.cimne.upc.es/cfsi/frontal/doc/ppt/11.pdf>
- Object Management Group. (29 de Mayo de 2014). *Unified Modeling Language*. Obtenido de <http://www.uml.org/>
- Open Source Community. (29 de Mayo de 2014). *Open Source Initiative*. Obtenido de <http://opensource.org/>
- Ortega Chasi, P. M. (2012). *La multimedia como herramienta de apoyo en la investigación de Psicología*. Cuenca, Azuay, Cuenca.
- Parasuraman, R. (1984). Sustained attention in detection and discrimination. En R. Parasuraman, & D. Davies, *Varieties of attention* (págs. 243-271). New York: Academic Press.
- PHP Documentation Group. (28 de Abril de 2014). *PHP*. (P. Olson, Editor) Obtenido de <http://www.php.net/manual/es/intro-what-is.php>
- Quintana, G., Marqués, M., Aliaga, J. L., & Arambur, M. J. (2008). *Aprende SQL*. Universitat Jaume I. Obtenido de <http://goo.gl/VALcNy>
- Robledo Sacristan, C., & Robledo Fernández, D. (2012). *Programación en Android* (Susana Perez Marín ed.). España. Recuperado el 24 de Abril de 2014, de <http://goo.gl/uJfoIv>

- The Official CAPTCHA Site*. (29 de Mayo de 2014). Obtenido de <http://www.captcha.net/>
- Thurstone, L. L. (2013). *Manual Formas Idénticas-R* (Sexta ed.). Madrid: TEA Ediciones. Obtenido de http://www.web.teaediciones.com/Ejemplos/Formas_identicas-R%20EXTRACTO.pdf
- Thurstone, L., & Yela, M. (2009). *Percepción de Diferencias (caras) Manual*. (Décima ed.). Madrid, España: TEA Ediciones, S.A.
- Thurstone, L., & Yela, M. (2012). *Caras-R Test de Percepción de Diferencias - Revisado*. (Undécima ed.). Madrid: TEA Ediciones S. A. Obtenido de http://www.web.teaediciones.com/Ejemplos/CARAS-R%20Manual_2012.pdf
- Tomas, J., & Almenara, J. (s.f.). *Master en Paidopsiquiatría*. Recuperado el 22 de Junio de 2013, de http://www.paidopsiquiatria.cat/files/Evaluacion_Atencion.pdf

ANEXOS

TUTORIAL DE INICIACIÓN EN PROGRAMACIÓN ANDROID

www.olimpiadasandroid.com

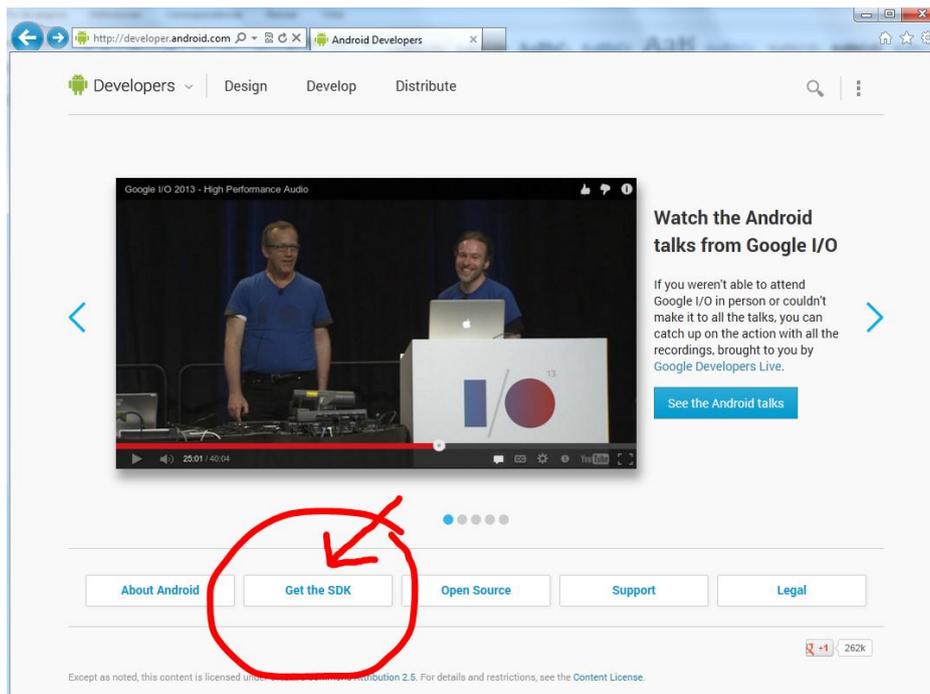
¿Quieres aprender Android desde cero y participar en las Olimpiadas Android?

¡Es muy fácil! Nosotros te decimos cómo, paso a paso.

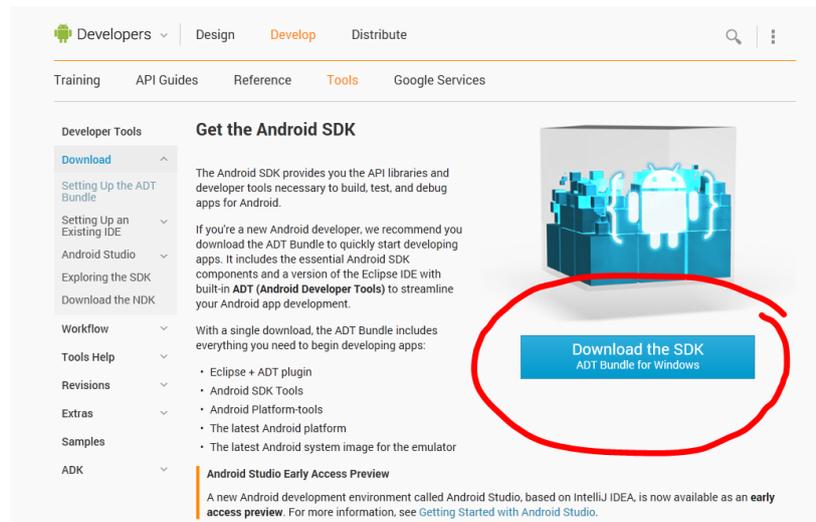
Pasarás un buen rato, aprenderás mucho y además podrás ganar fantásticos premios. Todo lo que necesitas para aprender y ganar está en este sencillo y corto tutorial.

Sigue exactamente los siguientes pasos para empezar a programar tu primera aplicación Android:

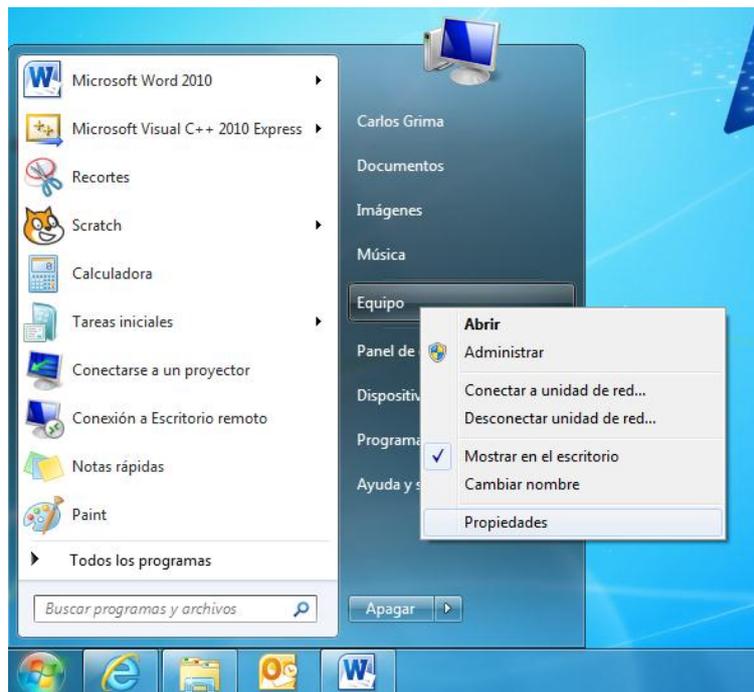
1. Acude a <http://developer.android.com> Durante el concurso podrás acceder a todos los contenidos de esta página.
2. Pulsa en "Get the SDK". El SDK es el compilador para programar en Android e incluye también el simulador de cualquier teléfono Android.

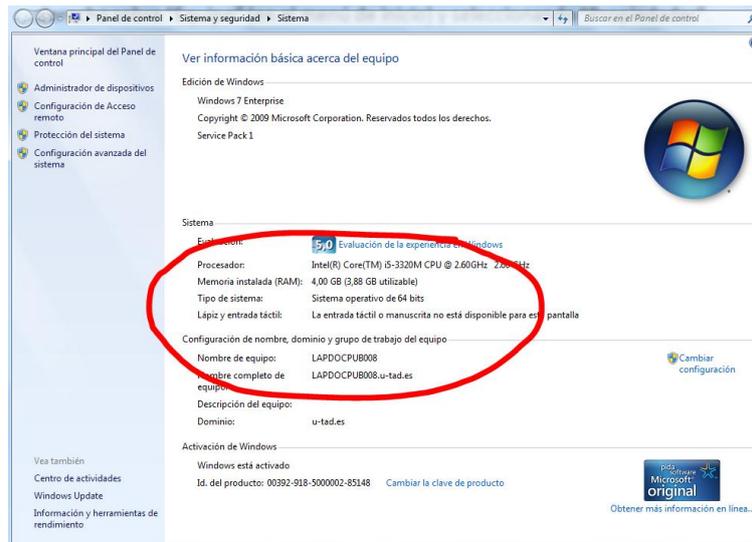


3. En la siguiente pantalla, pulsa en el botón “Download the SDK. ADT Bundle for Windows”. En el evento utilizaremos Windows 7.

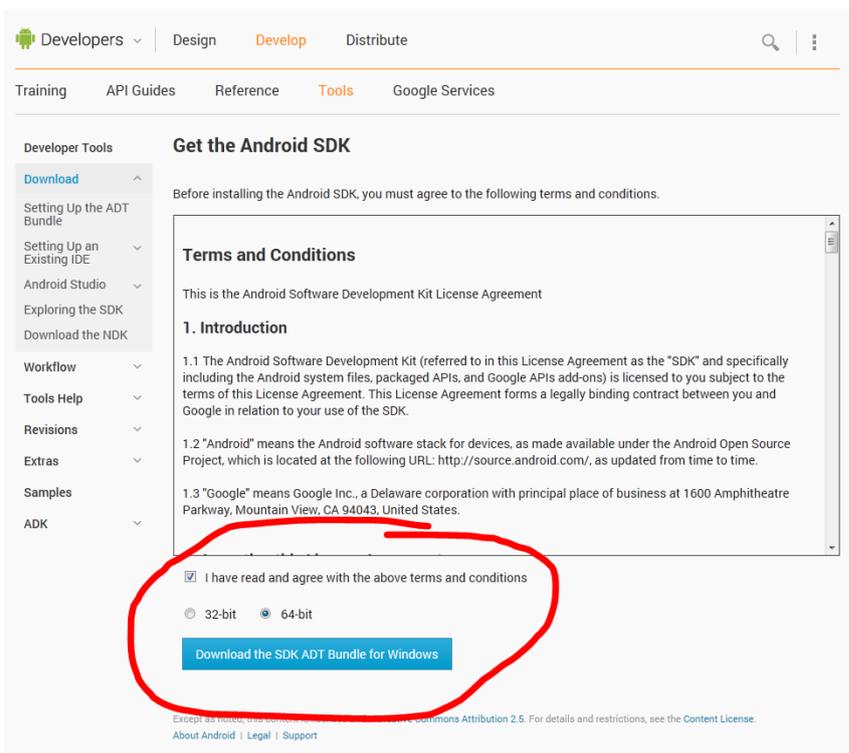


4. Si no lo sabes, averigua si tu sistema es de 32 o de 64 bits pulsando con el botón derecho sobre “Equipo” (en el menú de inicio) y seleccionando “Propiedades”.





5. En la siguiente pantalla después del paso 3, acepta las condiciones, selecciona si tu sistema es 32 ó 64 bits, y pulsa en el botón “Download the SDK ADT Bundle for Windows”.



6. Una vez descargado el archivo, descomprímelo en la carpeta que quieras.
7. Ahora necesitamos descargar e instalar el JDK, que es el compilador para programar en el lenguaje Java. La programación en Android suele hacerse casi siempre sobre el lenguaje Java. Acude a <http://java.oracle.com>

8. Pulsa en “Java SE”, dentro del apartado “Top downloads”.

The screenshot shows the Oracle Java website's 'Software Downloads' page. At the top, there's a navigation bar with 'ORACLE' and various links like 'Sign In/Register for Account', 'Help', 'Select Country/Region', 'Communities', 'I am a...', and 'I want to...'. Below this is a secondary navigation bar with 'PRODUCTS AND SERVICES', 'SOLUTIONS', 'DOWNLOADS', 'STORE', 'SUPPORT', 'TRAINING', 'PARTNERS', and 'ABOUT'. The main content area features a 'Virtual Developer Days: June 19 and 25' banner. To the right, the 'Software Downloads' section is highlighted with a red border. Within this section, the 'Top Downloads' list is circled in red, showing 'Java SE' as the top item, followed by 'Java EE and GlassFish', 'JavaFX', 'Java ME', 'JDeveloper 11g and ADF', 'Enterprise Pack for Eclipse', 'NetBeans IDE', and 'Pre-Built VM for Java Devs'. Other sections include 'New Downloads' and 'What's New'.

9. En la siguiente pantalla, descarga el JDK pulsando en el botón “JDK Download”.

The screenshot shows the Oracle Java website's 'Java SE Downloads' page. The navigation bar is similar to the previous screenshot. The main content area is titled 'Java SE Downloads' and has tabs for 'Overview', 'Downloads', 'Documentation', 'Community', 'Technologies', and 'Training'. The 'Downloads' tab is active. Below the tabs, there are three download buttons: 'Java Platform (JDK) 7u21', 'JavaFX 2.2.21', and 'JDK 7 + NetBeans'. The 'JDK' button is circled in red. Below these buttons, there's a section for 'Java Platform, Standard Edition' with a 'Java SE 7u21' release note. A 'Which Java package do I need?' section lists 'JDK', 'Server JRE', and 'JRE' with their respective descriptions. At the bottom, there are three columns for 'JDK 7 Docs', 'Server JRE 7 Docs', and 'JRE 7 Docs', each with a 'DOWNLOAD' button. The 'JDK' button is circled in red. On the right side, there's a 'Java SDKs and Tools' section with a list of links, and a 'Virtual Developer Day' banner.

10. En la siguiente pantalla, acude al bloque gris “Java SE Development Kit 7u21” (o la última versión que aparezca), acepta la licencia y pulsa sobre la descarga para “Windows x86” o para “Windows x64”, según tu sistema sea de 32 o de 64 bits respectivamente.

The screenshot shows the Oracle Java SE Development Kit 7u21 Downloads page. The page is titled "Java SE Development Kit 7 Downloads" and includes a table of download links for various operating systems and architectures. A red circle highlights the "Accept License Agreement" radio button, and another red circle highlights the "Windows x86" download link.

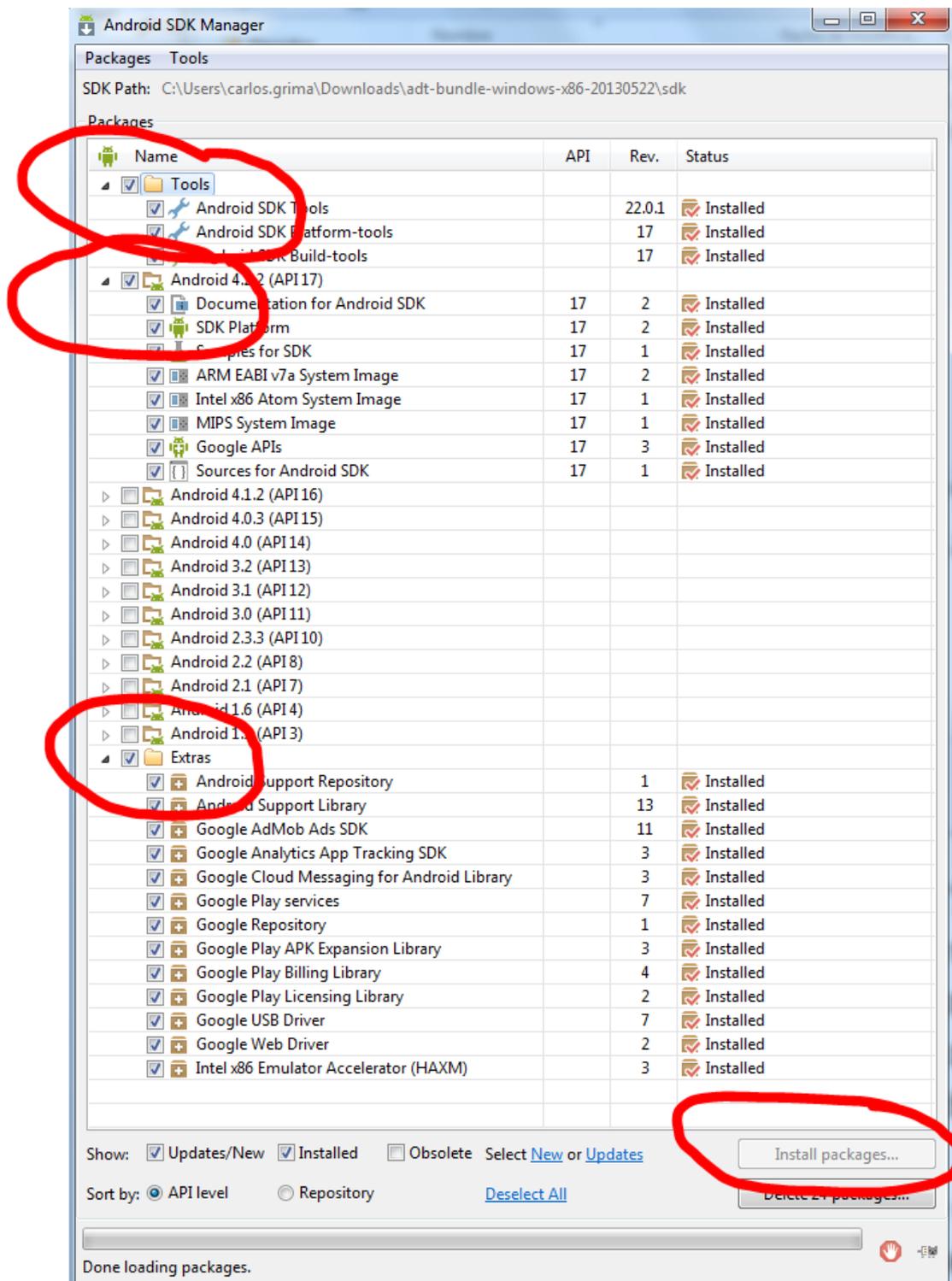
Product / File Description	File Size	Download
Linux ARM v6/v7 Soft Float ABI	65.09 MB	jdk-7u21-linux-arm-sfp.tar.gz
Linux x86	80.35 MB	jdk-7u21-linux-i586.rpm
Linux x86	93.06 MB	jdk-7u21-linux-i586.tar.gz
Linux x64	81.43 MB	jdk-7u21-linux-x64.rpm
Linux x64	91.81 MB	jdk-7u21-linux-x64.tar.gz
Mac OS X x64	144.18 MB	jdk-7u21-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.84 MB	jdk-7u21-solaris-i586.tar.Z
Solaris x86	92.08 MB	jdk-7u21-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.67 MB	jdk-7u21-solaris-x64.tar.Z
Solaris x64	15.02 MB	jdk-7u21-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	136.09 MB	jdk-7u21-solaris-sparc.tar.Z
Solaris SPARC	95.44 MB	jdk-7u21-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	26.63 MB	jdk-7u21-solaris-sparc64.tar.Z
Solaris SPARC 64-bit	17.58 MB	jdk-7u21-solaris-sparc64.tar.gz
Windows x86	88.98 MB	jdk-7u21-windows-i586.exe
Windows x64	90.57 MB	jdk-7u21-windows-x64.exe

11. Una vez descargado, instálalo.

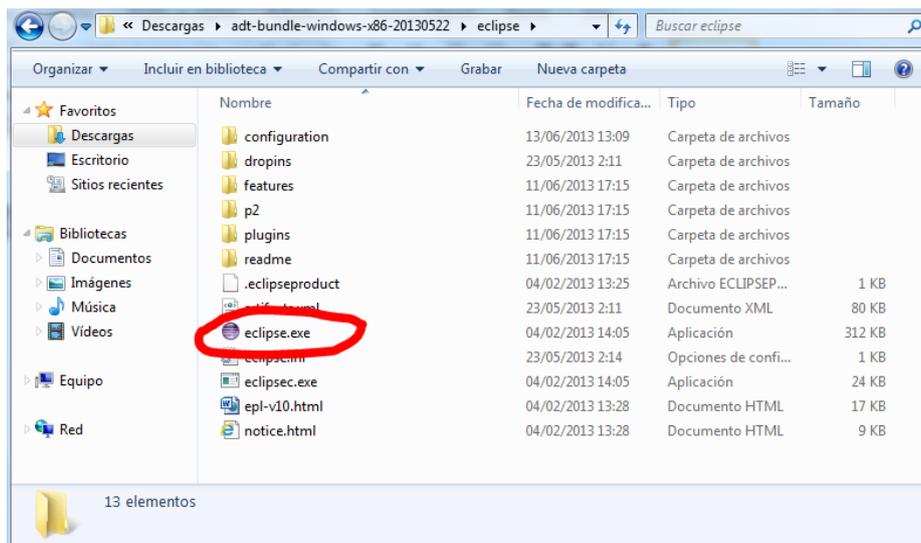
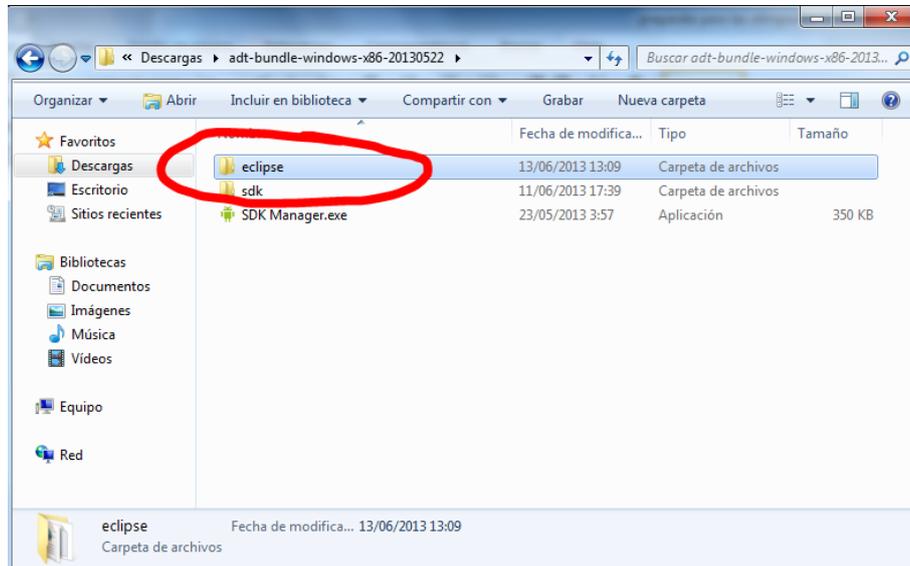
12. Una vez instalado el JDK, entra en la carpeta en donde descomprimiste el SDK de Android y ejecuta el archivo “SDK Manager”, haciendo doble click sobre él. Tardará unos segundos en abrirse.

The screenshot shows a Windows Explorer window displaying the contents of a folder named "adt-bundle-windows-x86-20130522". The folder contains three items: "eclipse", "sdk", and "SDK Manager.exe". The "SDK Manager.exe" file is circled in red.

13. Selecciona todas las "Tools", todo el "Android 4.2.2 (API17)" (o la última versión disponible) y todos los "Extras". Pulsa el botón "Install packages".

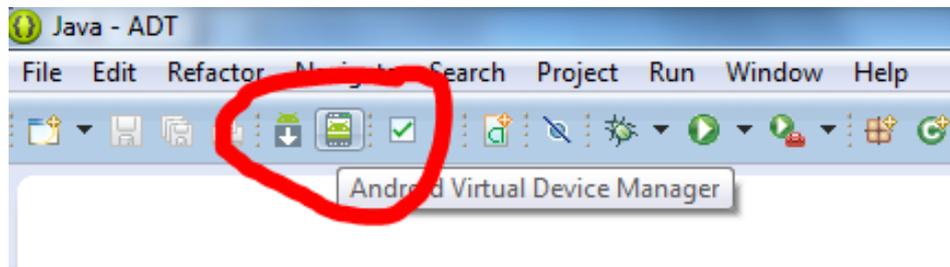


14. Una vez instalado todo, ya puedes ejecutar “Eclipse”, que es el entorno de desarrollo para Android. Dentro de la carpeta en donde descomprimiste el SDK, entra en la subcarpeta “Eclipse” y haz doble click sobre el archivo que se llama “Eclipse” para ejecutarlo.



15. Acepta el workspace que te sugiere.

16. Una vez iniciado todo, pulsa sobre el botón “Android Virtual Device Manager” y luego en “new” para crear tu dispositivo móvil virtual para probar tus aplicaciones. Una vez creado, inícialo pulsando en “start”.



17. ¡Ya puedes empezar a programar tu primera aplicación Android! Acude a <http://developer.android.com> de nuevo. Pulsa arriba en “Develop” y, en la siguiente pantalla, arriba en “Training”. Te aparecerá el tutorial oficial de Android con todos los pasos, desde los más sencillos (crear tu primera aplicación) hasta los más complicados. ¡Todo lo que necesitas para ganar el concurso está ahí!

A screenshot of the Android Developer website. The top navigation bar includes 'Developers', 'Design', 'Develop', and 'Distribute'. Below this, there are tabs for 'Training', 'API Guides', 'Reference', 'Tools', and 'Google Services'. The 'Training' section is expanded, showing a list of topics on the left and a main content area on the right. The main content area is titled 'Getting Started' and includes a welcome message, a list of links for 'Building Your First App', 'Managing the Activity Lifecycle', 'Supporting Different Devices', and 'Building a Dynamic UI with Fragments'. Each section includes a brief description and a list of related links.

Estructura de la Base de Datos en MySQL utilizado en el Test.

CREAR TABLA GPSI_AGENDA

```
CREATE TABLE IF NOT EXISTS `GPSI_AGENDA` (  
  `AGE_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `USU_CI` varchar(10) NOT NULL,  
  `AGE_FECHA` datetime NOT NULL,  
  `AGE_HORA` datetime NOT NULL,  
  `AGE_OBSERVACION` text,  
  `AGE_ESTADO` char(1) NOT NULL,  
  `AGE_ACCION` char(1) NOT NULL,  
  `AGE_CREATED_AT` datetime NOT NULL,  
  `AGE_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  PRIMARY KEY (`AGE_ID`),  
  KEY `AGENDA_USUARIO_FK` (`USU_CI`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA GPSI_FICHA

```
CREATE TABLE IF NOT EXISTS `GPSI_FICHA` (  
  `PSI_NUMERO` int(11) NOT NULL AUTO_INCREMENT,  
  `USU_CI` varchar(10) NOT NULL,  
  `PSI_CEDULA` varchar(10) DEFAULT NULL,  
  `PSI_NOMBRE1` varchar(50) NOT NULL,  
  `PSI_NOMBRE2` varchar(50) DEFAULT NULL,  
  `PSI_APELLIDO1` varchar(50) NOT NULL,  
  `PSI_APELLIDO2` varchar(50) DEFAULT NULL,  
  `PSI_DIRECCION` varchar(50) DEFAULT NULL,  
  `PSI_TELEFONO` varchar(20) DEFAULT NULL,  
  `PSI_CELULAR` varchar(20) DEFAULT NULL,  
  `PSI_EMAIL` varchar(50) DEFAULT NULL,  
  `PSI_LUGARNAC` varchar(50) DEFAULT NULL,  
  `PSI_FECNACIMIENTO` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',  
  `PSI_SEXO` char(1) NOT NULL,  
  `PSI_NACIONALIDAD` varchar(20) DEFAULT NULL,  
  `ESC_CODIGO` varchar(10) CHARACTER SET latin1 COLLATE latin1_spanish_ci  
  NOT NULL,  
  `PSI_NIVEL_ESCOLAR` int(2) NOT NULL,  
  `PAR_CODIGO` varchar(6) CHARACTER SET latin1 COLLATE latin1_spanish_ci  
  NOT NULL,  
  `PSI_NOMPADRE` varchar(100) DEFAULT NULL,  
  `PSI_PROPADRE` varchar(50) DEFAULT NULL,  
  `PSI_TELEFPADRE` varchar(20) DEFAULT NULL,  
  `PSI_NACPADRE` varchar(20) DEFAULT NULL,  
  `PSI_NOMMADRE` varchar(100) DEFAULT NULL,  
  `PSI_PROMADRE` varchar(50) DEFAULT NULL,  
  `PSI_TELEFMADRE` varchar(20) DEFAULT NULL,
```

`PSI_NACMADRE` varchar(20) DEFAULT NULL,
 `PSI_NUMHERMANOS` int(2) DEFAULT '0',
 `PSI_FECHA` datetime DEFAULT '0000-00-00 00:00:00',
 `PSI_REMITIDO` varchar(255) DEFAULT NULL,
 `PSI_OBSERVACION` text,
 `PSI_OCURRENCIA` varchar(255) DEFAULT NULL,
 `PSI_DESDE` varchar(255) DEFAULT NULL,
 `PSI_ATRIBUYE` varchar(255) DEFAULT NULL,
 `PSI_FUENTE` varchar(255) DEFAULT NULL,
 `PSI_ENFERMEDAD_ACTUAL` varchar(255) DEFAULT NULL,
 `PSI_FECHA_TRAT_BIOLOGICO` datetime DEFAULT '0000-00-00 00:00:00',
 `PSI_TRATAMIENTO_BIOLOGICO` varchar(255) DEFAULT NULL,
 `PSI_FECHA_TRAT_PSICOLOGICO` datetime DEFAULT '0000-00-00 00:00:00',
 `PSI_TRATAMIENTO_PSICOLOGICO` varchar(255) DEFAULT NULL,
 `PSI_PARTO` varchar(255) DEFAULT NULL,
 `PSI_LACTANCIA` varchar(255) DEFAULT NULL,
 `PSI_PSICOMOTOR` varchar(255) DEFAULT NULL,
 `PSI_ENFERMEDAD_SOMATICA` varchar(255) DEFAULT NULL,
 `PSI_SIGNOS_PSICOPATOLOGICOS` varchar(255) DEFAULT NULL,
 `PSI_RELACION_PADRES` varchar(255) DEFAULT NULL,
 `PSI_RELACION_HERMANOS` varchar(255) DEFAULT NULL,
 `PSI_OTROS_INFANCIA` varchar(255) DEFAULT NULL,
 `PSI_OTROS_PREESCOLAR` varchar(255) DEFAULT NULL,
 `PSI_EDUCACION_ESCOLAR` varchar(255) DEFAULT NULL,
 `PSI_OTROS_ESCOLAR` varchar(255) DEFAULT NULL,
 `PSI_ADOLESCENCIA` varchar(255) DEFAULT NULL,
 `PSI_ADAPTACION_HETEROSEXUAL` varchar(255) DEFAULT NULL,
 `PSI_VIDA_SOCIAL` varchar(255) DEFAULT NULL,
 `PSI_CLASE_SOCIOECONOMICA` varchar(255) DEFAULT NULL,
 `PSI_USO_SUSTANCIAS` varchar(255) DEFAULT NULL,
 `PSI_PATOLOGIA_ANTERIOR` varchar(255) DEFAULT NULL,
 `PSI_ANTECEDENTES_FAMILIARES` text,
 `PSI_INSPECCION_GENERAL` varchar(255) DEFAULT NULL,
 `PSI_FORMA_RELACION` varchar(255) DEFAULT NULL,
 `PSI_EXPRESION LENGUAJE` varchar(255) DEFAULT NULL,
 `PSI_ESTADO_CONCIENCIA` varchar(255) DEFAULT NULL,
 `PSI_ATENCION` varchar(255) DEFAULT NULL,
 `PSI_CONDUCTA_MOTORA` varchar(255) DEFAULT NULL,
 `PSI_COMPORTAMIENTO` varchar(255) DEFAULT NULL,
 `PSI_ALTERACION_SUENIO` varchar(255) DEFAULT NULL,
 `PSI_AFECTIVIDAD` varchar(255) DEFAULT NULL,
 `PSI_SENSOPERCEPCION` varchar(255) DEFAULT NULL,
 `PSI_PENSAMIENTO` varchar(255) DEFAULT NULL,
 `PSI_EXPRESION` varchar(255) DEFAULT NULL,
 `PSI_JUICIO` varchar(255) DEFAULT NULL,
 `PSI_ESTIMACION_INTELIGENCIA` varchar(255) DEFAULT NULL,
 `PSI_ORIENTACION` varchar(255) DEFAULT NULL,
 `PSI_MEMORIA` varchar(255) DEFAULT NULL,
 `PSI_OTRAS_FUNCIONES_INT` varchar(255) DEFAULT NULL,
 `PSI_VOLUNTAD` varchar(255) DEFAULT NULL,

```

`PSI_ACTIVIDAD` varchar(255) DEFAULT NULL,
`PSI_INSTINTIVIDAD` varchar(255) DEFAULT NULL,
`PSI_CONCLUSIONES` text,
`PSI_RECOMENDACIONES` text,
`PSI_ESTADO` char(1) NOT NULL,
`PSI_ACCION` char(1) NOT NULL,
`PSI_CREATED_AT` datetime NOT NULL,
`PSI_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',
`PSI_DISPOSITIVO` varchar(50) DEFAULT NULL,
PRIMARY KEY (`PSI_NUMERO`),
KEY `FICHA_USUARIO_FK` (`USU_CI`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

CREAR TABLA GPSI_INTERCONSULTA

```

CREATE TABLE IF NOT EXISTS `GPSI_INTERCONSULTA` (
  `INT_NUMERO` int(11) NOT NULL AUTO_INCREMENT,
  `PSI_NUMERO` int(11) NOT NULL,
  `MOT_CODIGO` int(11) NOT NULL,
  `INT_FECHA` datetime DEFAULT '0000-00-00 00:00:00',
  `INT_ESTADO_PSI` text,
  `INT_TRATAMIENTO` text,
  `INT_OBSERVACION` text,
  `INT_ESTADO_FICHA` text,
  `INT_COMENTARIO` text,
  `INT_ESTADO` char(1) NOT NULL,
  `INT_ACCION` char(1) NOT NULL,
  `INT_CREATED_AT` datetime NOT NULL,
  `INT_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',
  `INT_DISPOSITIVO` varchar(50) DEFAULT NULL,
  `USU_CI` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`INT_NUMERO`),
  KEY `INTERCON_FICHA_FK` (`PSI_NUMERO`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

CREA TABLA GPSI_MOTIVO_CONSULTA

```

CREATE TABLE IF NOT EXISTS `GPSI_MOTIVO_CONSULTA` (
  `MOT_CODIGO` int(11) NOT NULL AUTO_INCREMENT,
  `MOT_NOMBRE` varchar(100) NOT NULL,
  `MOT_ESTADO` char(1) NOT NULL,
  `MOT_ACCION` char(1) NOT NULL,
  `MOT_CREATED_AT` datetime NOT NULL,
  `MOT_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',
  `USU_CI` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`MOT_CODIGO`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

CREAR TABLA GSEG_ACCESO

```
CREATE TABLE IF NOT EXISTS `GSEG_ACCESO` (  
  `ACC_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `USU_CI` varchar(10) NOT NULL,  
  `ACC_FECHA` datetime NOT NULL,  
  `ACC_EXPLORADOR` varchar(50) DEFAULT NULL,  
  `ACC_EXPLORADOR_VERSION` varchar(20) NOT NULL,  
  `ACC_SISTEMA_OPERATIVO` varchar(20) NOT NULL,  
  `ACC_DISPOSITIVO` varchar(50) NOT NULL,  
  `ACC_IP` varchar(55) NOT NULL,  
  `ACC_CREATED_AT` datetime NOT NULL,  
  `ACC_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  PRIMARY KEY (`ACC_ID`),  
  KEY `ACCESO_USUARIO_FK` (`USU_CI`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA GSEG_CANTONES

```
CREATE TABLE IF NOT EXISTS `GSEG_CANTONES` (  
  `CAN_CODIGO` varchar(4) COLLATE latin1_spanish_ci NOT NULL,  
  `CAN_NOMBRE` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `PRO_CODIGO` varchar(2) COLLATE latin1_spanish_ci NOT NULL,  
  `CAN_ESTADO` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `CAN_ACCION` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `CAN_CREATED_AT` datetime NOT NULL,  
  `CAN_UPDATE_AT` datetime DEFAULT NULL,  
  `USU_CI` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
  PRIMARY KEY (`CAN_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

CREAR TABLA GSEG_ESCUELAS

```
CREATE TABLE IF NOT EXISTS `GSEG_ESCUELAS` (  
  `ESC_CODIGO` varchar(10) COLLATE latin1_spanish_ci NOT NULL,  
  `ESC_NOMBRE` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `PAR_CODIGO` varchar(6) COLLATE latin1_spanish_ci NOT NULL,  
  `ESC_ESTADO` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `ESC_ACCION` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `ESC_CREATED_AT` datetime NOT NULL,  
  `ESC_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
  KEY `ESC_CODIGO` (`ESC_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

CREAR TABLA GSEG_MARGARITA_SERIE

```
CREATE TABLE IF NOT EXISTS `GSEG_MARGARITA_SERIE` (  
  `SER_ID` int(11) NOT NULL,  
  `SER_EXPRESION` char(1) COLLATE latin1_spanish_ci NOT NULL COMMENT  
'A=alegre; T=triste; E=enojado; M=miedo',  
  `SER_CREATED_AT` datetime NOT NULL,  
  `SER_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
  PRIMARY KEY (`SER_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

CREAR TABLA GSEG_MENU

```
CREATE TABLE IF NOT EXISTS `GSEG_MENU` (  
  `MEN_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `MEN_NOMBRE` varchar(50) DEFAULT NULL,  
  `MEN_FORMULARIO` varchar(50) DEFAULT NULL,  
  `MEN_NIVEL` char(1) DEFAULT NULL,  
  `MEN_PADRE` int(11) DEFAULT '0' COMMENT 'Nodo padre',  
  `MEN_ESTADO` char(1) NOT NULL,  
  `MEN_ACCION` char(1) NOT NULL,  
  `MEN_CREATED_AT` datetime NOT NULL,  
  `MEN_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`MEN_CODIGO`),  
  KEY `MEN_PADRE` (`MEN_PADRE`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA GSEG_MENU_PANEL

```
CREATE TABLE IF NOT EXISTS `GSEG_MENU_PAPEL` (  
  `MEN_CODIGO` int(11) NOT NULL,  
  `PAP_CODIGO` int(11) NOT NULL,  
  `MPA_AGREGAR` char(1) NOT NULL,  
  `MPA_MODIFICAR` char(1) NOT NULL,  
  `MPA_ELIMINAR` char(1) NOT NULL,  
  `MPA_IMPRIMIR` char(1) NOT NULL,  
  `MPA_BUSCAR` char(1) NOT NULL,  
  `MPA_DESPLAZAR` char(1) NOT NULL,  
  `MPA_ESTADO` char(1) NOT NULL,  
  `MPA_ACCION` char(1) NOT NULL,  
  `MPA_CREATED_AT` datetime NOT NULL,  
  `MPA_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) DEFAULT NULL,  
  KEY `MENPAP_MENU_FK` (`MEN_CODIGO`),  
  KEY `MENPAP_PAPEL_FK` (`PAP_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

CREAR TABLA GSEG_PAPEL

```
CREATE TABLE IF NOT EXISTS `GSEG_PAPEL` (  
  `PAP_CODIGO` int(11) NOT NULL AUTO_INCREMENT,  
  `PAP_NOMBRE` varchar(100) NOT NULL,  
  `PAP_ESTADO` char(1) NOT NULL,  
  `PAP_ACCION` char(1) NOT NULL,  
  `PAP_CREATED_AT` datetime NOT NULL,  
  `PAP_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`PAP_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA GSEG_PARROQUIAS

```
CREATE TABLE IF NOT EXISTS `GSEG_PARROQUIAS` (  
  `PAR_CODIGO` varchar(6) COLLATE latin1_spanish_ci NOT NULL,  
  `PAR_NOMBRE` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `CAN_CODIGO` varchar(4) COLLATE latin1_spanish_ci NOT NULL,  
  `PRO_CODIGO` varchar(2) COLLATE latin1_spanish_ci NOT NULL,  
  `PAR_ESTADO` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `PAR_ACCION` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `PAR_CREATED_AT` datetime NOT NULL,  
  `PAR_UPDATE_AT` datetime DEFAULT NULL,  
  `USU_CI` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
  PRIMARY KEY (`PAR_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

CREAR TABLA GSEG_PROVINCIAS

```
CREATE TABLE IF NOT EXISTS `GSEG_PROVINCIAS` (  
  `PRO_CODIGO` varchar(2) COLLATE latin1_spanish_ci NOT NULL,  
  `PRO_NOMBRE` varchar(100) COLLATE latin1_spanish_ci NOT NULL,  
  `PRO_ESTADO` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `PRO_ACCION` char(1) COLLATE latin1_spanish_ci NOT NULL,  
  `PRO_CREATED_AT` datetime NOT NULL,  
  `PRO_UPDATE_AT` datetime DEFAULT NULL,  
  `USU_CI` varchar(10) COLLATE latin1_spanish_ci DEFAULT NULL,  
  PRIMARY KEY (`PRO_CODIGO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

CREAR TABLA GSEG_REG_MOVIMIENTOS

```
CREATE TABLE IF NOT EXISTS `GSEG_REG_MOVIMIENTOS` (  
  `RMOV_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `RMOV_SENTENCIA` text NOT NULL,  
  `RMOV_NOM_TABLA` varchar(100) NOT NULL,
```

```
PRIMARY KEY (`RMOV_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA GSEG_USUARIO

```
CREATE TABLE IF NOT EXISTS `GSEG_USUARIO` (  
  `USU_CI` varchar(10) NOT NULL,  
  `PAP_CODIGO` int(11) NOT NULL,  
  `USU_NOMBRE_USUARIO` varchar(50) NOT NULL,  
  `USU_CLAVE` varchar(50) NOT NULL,  
  `USU_TIEMPO_CITA` int(11) DEFAULT '30' COMMENT 'En minutos',  
  `USU_ESTADO` char(1) NOT NULL,  
  `USU_ACCION` char(1) NOT NULL,  
  `USU_NOMBRE1` varchar(50) NOT NULL,  
  `USU_NOMBRE2` varchar(50) DEFAULT '',  
  `USU_APELLIDO1` varchar(50) NOT NULL,  
  `USU_APELLIDO2` varchar(50) DEFAULT '',  
  `USU_EMAIL` varchar(50) DEFAULT '',  
  `USU_CREATED_AT` datetime NOT NULL,  
  `USU_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  PRIMARY KEY (`USU_CI`),  
  UNIQUE KEY `USU_NOMBRE_USUARIO` (`USU_NOMBRE_USUARIO`),  
  KEY `USUARIO_PAPEL_FK` (`PAP_CODIGO`),  
  KEY `USU_CLAVE` (`USU_CLAVE`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

CREAR TABLA TPA_APLICACION

```
CREATE TABLE IF NOT EXISTS `TPA_APLICACION` (  
  `APLI_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `TEST_ID` int(11) NOT NULL,  
  `PSI_NUMERO` int(11) NOT NULL,  
  `APLI_FECHA` datetime DEFAULT '0000-00-00 00:00:00',  
  `APLI_EQUIVOCACIONES` int(11) NOT NULL DEFAULT '0',  
  `APLI_ESTADO` char(1) NOT NULL,  
  `APLI_ACCION` char(1) NOT NULL,  
  `APLI_CREATED_AT` datetime NOT NULL,  
  `APLI_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `APLI_DISPOSITIVO` varchar(50) DEFAULT NULL,  
  `USU_CI` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`APLI_ID`),  
  KEY `APLICACION_TEST_FK` (`TEST_ID`),  
  KEY `APLICACION_FICHA_FK` (`PSI_NUMERO`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA TPA_APLICACION_DET

```
CREATE TABLE IF NOT EXISTS `TPA_APLICACION_DET` (  
  `DAPLI_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `PREG_ID` int(11) NOT NULL,  
  `RESP_ID` int(11) DEFAULT NULL,  
  `RESP_CORRECTA` char(1) DEFAULT NULL COMMENT 'S: correcta -  
N:incorrecta',  
  `APLI_ID` int(11) NOT NULL,  
  `DAPLI_TIEMPO` float(6,3) DEFAULT NULL COMMENT 'En segundos',  
  `DAPLI_ESTADO` char(1) NOT NULL,  
  `DAPLI_ACCION` char(1) NOT NULL,  
  `DAPLI_CREATED_AT` datetime NOT NULL,  
  `DAPLI_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `DAPLI_DISPOSITIVO` varchar(50) DEFAULT NULL,  
  `USU_CI` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`DAPLI_ID`),  
  KEY `APLIDET_PREGUNTA_FK` (`RESP_ID`),  
  KEY `APLIDET_APLICACION_FK` (`APLI_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA TPA_APLICACION_INT

```
CREATE TABLE IF NOT EXISTS `TPA_APLICACION_INT` (  
  `IAPLI_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `DINT_ID` int(11) NOT NULL,  
  `DINT_CORRECTA` char(1) NOT NULL,  
  `APLI_ID` int(11) NOT NULL,  
  `IAPLI_TIEMPO` float(6,3) NOT NULL COMMENT 'En segundos',  
  `IAPLI_ESTADO` char(1) NOT NULL,  
  `IAPLI_ACCION` char(1) NOT NULL,  
  `IAPLI_CREATED_AT` datetime NOT NULL,  
  `IAPLI_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `IAPLI_DISPOSITIVO` varchar(50) DEFAULT NULL,  
  `USU_CI` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`IAPLI_ID`),  
  KEY `APLIINT_INTERFERENCIA_FK` (`DINT_ID`),  
  KEY `APLIINT_APLICACION_FK` (`APLI_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA TPA_INTERFERENCIA

```
CREATE TABLE IF NOT EXISTS `TPA_INTERFERENCIA` (  
  `INT_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `INT_NOMBRE` varchar(50) NOT NULL,  
  `INT_DESCRIPCION` text,  
  `INT_TIPO` char(1) NOT NULL COMMENT 'C,S: Color, Sonido',  
  `INT_UBICACION` varchar(100) NOT NULL,
```

```

`INT_ESTADO` char(1) NOT NULL COMMENT 'N, A: Normal, Anulado',
`INT_ACCION` char(1) NOT NULL COMMENT 'N, M, A: Nuevo, Modificado,
Anulado',
`INT_CREATED_AT` datetime NOT NULL,
`INT_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',
`USU_CI` varchar(10) DEFAULT NULL,
PRIMARY KEY (`INT_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

CREAR TABLA TPA_INTERFERENCIA_DET

```

CREATE TABLE IF NOT EXISTS `TPA_INTERFERENCIA_DET` (
`DINT_ID` int(11) NOT NULL AUTO_INCREMENT,
`INT_ID` int(11) NOT NULL,
`DINT_ORDEN` int(11) NOT NULL,
`DINT_DESCRIPCION` varchar(100) NOT NULL,
`DINT_CORRECTA` char(1) NOT NULL COMMENT 'S, N: Si, No',
`DINT_UBICACION` varchar(100) NOT NULL,
`DINT_ESTADO` char(1) NOT NULL COMMENT 'N, A: Normal, Anulado',
`DINT_ACCION` char(1) NOT NULL COMMENT 'N, M, A: Nuevo, Modificado,
Anulado',
`DINT_CREATED_AT` datetime NOT NULL,
`DINT_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',
`USU_CI` varchar(10) DEFAULT NULL,
PRIMARY KEY (`DINT_ID`),
KEY `INTDET_INTERFERENCIA_FK` (`INT_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

CREAR TABLA TPA_PREGUNTA

```

CREATE TABLE IF NOT EXISTS `TPA_PREGUNTA` (
`PREG_ID` int(11) NOT NULL AUTO_INCREMENT,
`TEST_ID` int(11) NOT NULL,
`PREG_ORDEN` int(11) NOT NULL,
`PREG_EXPRESION` char(1) NOT NULL DEFAULT 'A' COMMENT 'A=alegre;
T=triste; E=enojado; M=miedo',
`PREG_ESTADO` char(1) NOT NULL COMMENT 'N, A: Normal, Anulado',
`PREG_ACCION` char(1) NOT NULL COMMENT 'N, M, A: Nuevo, Modificado,
Anulado',
`PREG_CREATED_AT` datetime NOT NULL,
`PREG_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',
`USU_CI` varchar(10) DEFAULT NULL,
PRIMARY KEY (`PREG_ID`),
KEY `PREGUNTA_TEST_FK` (`TEST_ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

CREAR TABLA TPA_RESPUESTA

```
CREATE TABLE IF NOT EXISTS `TPA_RESPUESTA` (  
  `RESP_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `TEST_ID` int(11) NOT NULL,  
  `PREG_ID` int(11) NOT NULL,  
  `RESP_ORDEN` int(11) NOT NULL,  
  `RESP_UBICACION` varchar(100) NOT NULL,  
  `RESP_CORRECTA` char(1) NOT NULL COMMENT 'S, N: Si, No',  
  `RESP_ESTADO` char(1) NOT NULL COMMENT 'N, A: Normal, Anulado',  
  `RESP_ACCION` char(1) NOT NULL COMMENT 'N, M, A: Nuevo, Modificado,  
Anulado',  
  `RESP_CREATED_AT` datetime NOT NULL,  
  `RESP_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) DEFAULT NULL,  
  PRIMARY KEY (`RESP_ID`),  
  KEY `RESPUESTA_PREGUNTA_FK` (`PREG_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

CREAR TABLA TPA_TEST

```
CREATE TABLE IF NOT EXISTS `TPA_TEST` (  
  `TEST_ID` int(11) NOT NULL AUTO_INCREMENT,  
  `TEST_NOMBRE` varchar(50) NOT NULL,  
  `TEST_DESCRIPCION` text NOT NULL,  
  `TEST_NUM_FILAS` int(1) NOT NULL,  
  `TEST_INSTRUCCIONES` int(2) NOT NULL,  
  `TEST_ESTADO` char(1) NOT NULL COMMENT 'N, A: Normal, Anulado',  
  `TEST_ACCION` char(1) NOT NULL COMMENT 'N, M, A: Nuevo, Modificado,  
Anulado',  
  `TEST_INTERFERENCIA` char(1) NOT NULL COMMENT 'S, N: Si, No',  
  `TEST_TIEMPO_INTER` int(3) NOT NULL,  
  `TEST_CREATED_AT` datetime NOT NULL,  
  `TEST_UPDATE_AT` datetime DEFAULT '0000-00-00 00:00:00',  
  `USU_CI` varchar(10) DEFAULT "",  
  PRIMARY KEY (`TEST_ID`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

TRIGGERS EN MYSQL

INSERTAR CANTONES

```
CREATE TRIGGER INSERT_CANTONES
AFTER INSERT ON GSEG_CANTONES
FOR EACH ROW
BEGIN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO GSEG_CANTONES
(CAN_CODIGO, CAN_NOMBRE, PRO_CODIGO, CAN_ESTADO, CAN_ACCION,
CAN_CREATED_AT, CAN_UPDATE_AT, USU_CI) VALUES (\',
NEW.CAN_CODIGO, \', \', NEW.CAN_NOMBRE, \', \', NEW.PRO_CODIGO, \', \',
NEW.CAN_ESTADO, \', \', NEW.CAN_ACCION, \', \',
NEW.CAN_CREATED_AT, \', \', COALESCE(NEW.CAN_UPDATE_AT, ''), \', \',
COALESCE(NEW.USU_CI, ''), \')', 'GSEG_CANTONES');
END
$$
```

ACTUALIZAR CANTONES

```
CREATE TRIGGER UPDATE_CANTONES
AFTER UPDATE ON GSEG_CANTONES
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.CAN_CODIGO, "") <>
COALESCE(NEW.CAN_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
CAN_CODIGO = \', COALESCE(NEW.CAN_CODIGO, ''), \\' WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, \'), 'GSEG_CANTONES');
        END IF;

    IF (COALESCE(OLD.CAN_NOMBRE, "") <>
COALESCE(NEW.CAN_NOMBRE, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
CAN_NOMBRE = \', COALESCE(NEW.CAN_NOMBRE, ''), \\' WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, \'), 'GSEG_CANTONES');
        END IF;

    IF (COALESCE(OLD.PRO_CODIGO, "") <>
COALESCE(NEW.PRO_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
PRO_CODIGO = \', COALESCE(NEW.PRO_CODIGO, ''), \\' WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, \'), 'GSEG_CANTONES');
        END IF;
```

```

        IF (COALESCE(OLD.CAN_ESTADO, "") <>
COALESCE(NEW.CAN_ESTADO, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
CAN_ESTADO = \', COALESCE(NEW.CAN_ESTADO, ''), '\ WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, '\'), 'GSEG_CANTONES');
        END IF;

```

```

        IF (COALESCE(OLD.CAN_ACCION, "") <>
COALESCE(NEW.CAN_ACCION, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
CAN_ACCION = \', COALESCE(NEW.CAN_ACCION, ''), '\ WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, '\'), 'GSEG_CANTONES');
        END IF;

```

```

        IF (COALESCE(OLD.CAN_CREATED_AT, "") <>
COALESCE(NEW.CAN_CREATED_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
CAN_CREATED_AT = \', COALESCE(NEW.CAN_CREATED_AT, ''), '\ WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, '\'), 'GSEG_CANTONES');
        END IF;

```

```

        IF (COALESCE(OLD.CAN_UPDATE_AT, "") <>
COALESCE(NEW.CAN_UPDATE_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
CAN_UPDATE_AT = \', COALESCE(NEW.CAN_UPDATE_AT, ''), '\ WHERE
CAN_CODIGO = \', OLD.CAN_CODIGO, '\'), 'GSEG_CANTONES');
        END IF;

```

```

        IF (COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_CANTONES SET
USU_CI = \', COALESCE(NEW.USU_CI, ''), '\ WHERE CAN_CODIGO = \',
OLD.CAN_CODIGO, '\'), 'GSEG_CANTONES');
        END IF;

```

```

END
$$

```

INSERTAR ESCUELAS

```

CREATE TRIGGER INSERT_ESCUELAS
AFTER INSERT ON GSEG_ESCUELAS
FOR EACH ROW
BEGIN

```

```

INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO GSEG_ESCUELAS
(ESC_CODIGO, ESC_NOMBRE, PAR_CODIGO, ESC_ESTADO, ESC_ACCION,
ESC_CREATED_AT, ESC_UPDATE_AT, USU_CI) VALUES (\',
NEW.ESC_CODIGO, \', \', NEW.ESC_NOMBRE, \', \', NEW.PAR_CODIGO, \', \',
NEW.ESC_ESTADO, \', \', NEW.ESC_ACCION, \', \', NEW.ESC_CREATED_AT,
\', \', COALESCE(NEW.ESC_UPDATE_AT, ''), \', \', COALESCE(NEW.USU_CI, ''),
\')'), 'GSEG_ESCUELAS');
END
$$

```

ACTUALIZAR ESCUELAS

```

CREATE TRIGGER UPDATE_ESCUELAS
AFTER UPDATE ON GSEG_ESCUELAS
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.ESC_CODIGO, "") <>
COALESCE(NEW.ESC_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
ESC_CODIGO = \', COALESCE(NEW.ESC_CODIGO, ''), '\ WHERE ESC_CODIGO
= \', OLD.ESC_CODIGO , '\'), 'GSEG_ESCUELAS');
        END IF;

    IF(COALESCE(OLD.ESC_NOMBRE, "") <>
COALESCE(NEW.ESC_NOMBRE, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
ESC_NOMBRE = \', COALESCE(NEW.ESC_NOMBRE, ''), '\ WHERE
ESC_CODIGO = \', OLD.ESC_CODIGO , '\'), 'GSEG_ESCUELAS');
        END IF;

    IF(COALESCE(OLD.PAR_CODIGO, "") <>
COALESCE(NEW.PAR_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
PAR_CODIGO = \', COALESCE(NEW.PAR_CODIGO, ''), '\ WHERE
ESC_CODIGO = \', OLD.ESC_CODIGO , '\'), 'GSEG_ESCUELAS');
        END IF;

    IF(COALESCE(OLD.ESC_ESTADO, "") <>
COALESCE(NEW.ESC_ESTADO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
ESC_ESTADO = \', COALESCE(NEW.ESC_ESTADO, ''), '\ WHERE ESC_CODIGO
= \', OLD.ESC_CODIGO , '\'), 'GSEG_ESCUELAS');
        END IF;

```

```

        IF(COALESCE(OLD.ESC_ACCION, "") <> COALESCE(NEW.ESC_ACCION,
"")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
ESC_ACCION = \', COALESCE(NEW.ESC_ACCION, ''), '\ WHERE ESC_CODIGO
= \', OLD.ESC_CODIGO , '\' ), 'GSEG_ESCUELAS');
        END IF;

        IF(COALESCE(OLD.ESC_CREATED_AT, "") <>
COALESCE(NEW.ESC_CREATED_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
ESC_CREATED_AT = \', COALESCE(NEW.ESC_CREATED_AT, ''), '\ WHERE
ESC_CODIGO = \', OLD.ESC_CODIGO , '\' ), 'GSEG_ESCUELAS');
        END IF;

        IF(COALESCE(OLD.ESC_UPDATE_AT, "") <>
COALESCE(NEW.ESC_UPDATE_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
ESC_UPDATE_AT = \', COALESCE(NEW.ESC_UPDATE_AT, ''), '\ WHERE
ESC_CODIGO = \', OLD.ESC_CODIGO , '\' ), 'GSEG_ESCUELAS');
        END IF;

        IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_ESCUELAS SET
USU_CI = \', COALESCE(NEW.USU_CI, ''), '\ WHERE ESC_CODIGO = \',
OLD.ESC_CODIGO , '\' ), 'GSEG_ESCUELAS');
        END IF;

END
$$

```

INSERTAR PARROQUIAS

```

CREATE TRIGGER INSERT_PARROQUIAS
AFTER INSERT ON GSEG_PARROQUIAS
FOR EACH ROW
BEGIN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO
GSEG_PARROQUIAS (PAR_CODIGO, PAR_NOMBRE, CAN_CODIGO,
PRO_CODIGO, PAR_ESTADO, PAR_ACCION, PAR_CREATED_AT,
PAR_UPDATE_AT, USU_CI) VALUES (\', NEW.PAR_CODIGO, '\, \',
NEW.PAR_NOMBRE, '\, \', NEW.CAN_CODIGO, '\, \', NEW.PRO_CODIGO, '\, \',
NEW.PAR_ESTADO, '\, \', NEW.PAR_ACCION, '\, \', NEW.PAR_CREATED_AT,
'\, \', COALESCE(NEW.PAR_UPDATE_AT, ''), '\, \', COALESCE(NEW.USU_CI,
''), '\' )', 'GSEG_PARROQUIAS' );

```

END
\$\$

ACTUALIZAR PARROQUIAS

```
CREATE TRIGGER UPDATE_PARROQUIAS
AFTER UPDATE ON GSEG_PARROQUIAS
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.PAR_CODIGO, "") <>
    COALESCE(NEW.PAR_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
        RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
        PAR_CODIGO = \'', COALESCE(NEW.PAR_CODIGO, ''), '\ ' WHERE
        PAR_CODIGO = \'', OLD.PAR_CODIGO , '\ ' ), 'GSEG_PARROQUIAS');
        END IF;

    IF(COALESCE(OLD.PAR_NOMBRE, "") <>
    COALESCE(NEW.PAR_NOMBRE, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
        RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
        PAR_NOMBRE = \'', COALESCE(NEW.PAR_NOMBRE, ''), '\ ' WHERE
        PAR_CODIGO = \'', OLD.PAR_CODIGO , '\ ' ), 'GSEG_PARROQUIAS');
        END IF;

    IF(COALESCE(OLD.CAN_CODIGO, "") <>
    COALESCE(NEW.CAN_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
        RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
        CAN_CODIGO = \'', COALESCE(NEW.CAN_CODIGO, ''), '\ ' WHERE
        PAR_CODIGO = \'', OLD.PAR_CODIGO , '\ ' ), 'GSEG_PARROQUIAS');
        END IF;

    IF(COALESCE(OLD.PRO_CODIGO, "") <>
    COALESCE(NEW.PRO_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
        RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
        PRO_CODIGO = \'', COALESCE(NEW.PRO_CODIGO, ''), '\ ' WHERE
        PAR_CODIGO = \'', OLD.PAR_CODIGO , '\ ' ), 'GSEG_PARROQUIAS');
        END IF;

    IF(COALESCE(OLD.PAR_ESTADO, "") <>
    COALESCE(NEW.PAR_ESTADO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
        RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
        PAR_ESTADO = \'', COALESCE(NEW.PAR_ESTADO, ''), '\ ' WHERE
        PAR_CODIGO = \'', OLD.PAR_CODIGO , '\ ' ), 'GSEG_PARROQUIAS');
        END IF;
```

```

        IF(COALESCE(OLD.PAR_ACCION, "") <>
        COALESCE(NEW.PAR_ACCION, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
            PAR_ACCION = \'', COALESCE(NEW.PAR_ACCION, ''), '\ ' WHERE
            PAR_CODIGO = \'', OLD.PAR_CODIGO, '\ ' ), 'GSEG_PARROQUIAS');
        END IF;
    
```

```

        IF(COALESCE(OLD.PAR_CREATED_AT, "") <>
        COALESCE(NEW.PAR_CREATED_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
            PAR_CREATED_AT = \'', COALESCE(NEW.PAR_CREATED_AT, ''), '\ ' WHERE
            PAR_CODIGO = \'', OLD.PAR_CODIGO, '\ ' ), 'GSEG_PARROQUIAS');
        END IF;
    
```

```

        IF(COALESCE(OLD.PAR_UPDATE_AT, "") <>
        COALESCE(NEW.PAR_UPDATE_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
            PAR_UPDATE_AT = \'', COALESCE(NEW.PAR_UPDATE_AT, ''), '\ ' WHERE
            PAR_CODIGO = \'', OLD.PAR_CODIGO, '\ ' ), 'GSEG_PARROQUIAS');
        END IF;
    
```

```

        IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PARROQUIAS SET
            USU_CI = \'', COALESCE(NEW.USU_CI, ''), '\ ' WHERE PAR_CODIGO = \'',
            OLD.PAR_CODIGO, '\ ' ), 'GSEG_PARROQUIAS');
        END IF;
    
```

```

END
$$
    
```

INSERTAR PROVINCIAS

```

CREATE TRIGGER INSERT_PROVINCIAS
AFTER INSERT ON GSEG_PROVINCIAS
FOR EACH ROW
BEGIN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
    RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO GSEG_PROVINCIAS
    (PRO_CODIGO, PRO_NOMBRE, PRO_ESTADO, PRO_ACCION,
    PRO_CREATED_AT, PRO_UPDATE_AT, USU_CI) VALUES (\',
    NEW.PRO_CODIGO, \', \', NEW.PRO_NOMBRE, \', \', NEW.PRO_ESTADO, \', \',
    NEW.PRO_ACCION, \', \', NEW.PRO_CREATED_AT, \', \',
    COALESCE(NEW.PRO_UPDATE_AT, ''), \', \', COALESCE(NEW.USU_CI, ''), '\ ' )
    ), 'GSEG_PROVINCIAS' );
    
```

END
\$\$

ACTUALIZAR PROVINCIAS

```
CREATE TRIGGER UPDATE_PROVINCIAS
AFTER UPDATE ON GSEG_PROVINCIAS
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.PRO_CODIGO, "") <>
COALESCE(NEW.PRO_CODIGO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
PRO_CODIGO = \'', COALESCE(NEW.PRO_CODIGO, ''), '\ ' WHERE
PRO_CODIGO = \'', OLD.PRO_CODIGO , '\ ' ), 'GSEG_PROVINCIAS');
        END IF;

    IF(COALESCE(OLD.PRO_NOMBRE, "") <>
COALESCE(NEW.PRO_NOMBRE, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
PRO_NOMBRE = \'', COALESCE(NEW.PRO_NOMBRE, ''), '\ ' WHERE
PRO_CODIGO = \'', OLD.PRO_CODIGO , '\ ' ), 'GSEG_PROVINCIAS');
        END IF;

    IF(COALESCE(OLD.PRO_ESTADO, "") <>
COALESCE(NEW.PRO_ESTADO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
PRO_ESTADO = \'', COALESCE(NEW.PRO_ESTADO, ''), '\ ' WHERE
PRO_CODIGO = \'', OLD.PRO_CODIGO , '\ ' ), 'GSEG_PROVINCIAS');
        END IF;

    IF(COALESCE(OLD.PRO_ACCION, "") <>
COALESCE(NEW.PRO_ACCION, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
PRO_ACCION = \'', COALESCE(NEW.PRO_ACCION, ''), '\ ' WHERE
PRO_CODIGO = \'', OLD.PRO_CODIGO , '\ ' ), 'GSEG_PROVINCIAS');
        END IF;

    IF(COALESCE(OLD.PRO_CREATED_AT, "") <>
COALESCE(NEW.PRO_CREATED_AT, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
PRO_CREATED_AT = \'', COALESCE(NEW.PRO_CREATED_AT, ''), '\ ' WHERE
PRO_CODIGO = \'', OLD.PRO_CODIGO , '\ ' ), 'GSEG_PROVINCIAS');
        END IF;
```

```

        IF(COALESCE(OLD.PRO_UPDATE_AT, "") <>
        COALESCE(NEW.PRO_UPDATE_AT, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
            PRO_UPDATE_AT = \', COALESCE(NEW.PRO_UPDATE_AT, ''), '\ WHERE
            PRO_CODIGO = \', OLD.PRO_CODIGO, '\'), 'GSEG_PROVINCIAS');
        END IF;

        IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_PROVINCIAS SET
            USU_CI = \', COALESCE(NEW.USU_CI, ''), '\ WHERE PRO_CODIGO = \',
            OLD.PRO_CODIGO, '\'), 'GSEG_PROVINCIAS');
        END IF;

    END
    $$

```

INSERTAR USUARIO

```

CREATE TRIGGER INSERT_USUARIO
AFTER INSERT ON GSEG_USUARIO
FOR EACH ROW
BEGIN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
    RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO GSEG_USUARIO
    (USU_CI, PAP_CODIGO, USU_NOMBRE_USUARIO, USU_CLAVE,
    USU_TIEMPO_CITA, USU_ESTADO, USU_ACCION, USU_NOMBRE1,
    USU_NOMBRE2, USU_APELLIDO1, USU_APELLIDO2, USU_EMAIL,
    USU_CREATED_AT, USU_UPDATE_AT) VALUES (\', NEW.USU_CI, \', \',
    COALESCE(NEW.PAP_CODIGO, 0), \', \',
    COALESCE(NEW.USU_NOMBRE_USUARIO, ''), \', \',
    COALESCE(NEW.USU_CLAVE, ''), \', \', COALESCE(NEW.USU_TIEMPO_CITA,
    0), \', \', COALESCE(NEW.USU_ESTADO, ''), \', \',
    COALESCE(NEW.USU_ACCION, ''), \', \', COALESCE(NEW.USU_NOMBRE1, ''),
    \', \', COALESCE(NEW.USU_NOMBRE2, ''), \', \',
    COALESCE(NEW.USU_APELLIDO1, ''), \', \',
    COALESCE(NEW.USU_APELLIDO2, ''), \', \', COALESCE(NEW.USU_EMAIL, ''),
    \', \', COALESCE(NEW.USU_CREATED_AT, ''), \', \',
    COALESCE(NEW.USU_UPDATE_AT, ''), \') '), 'GSEG_USUARIO' );
END
    $$

```

ACTUALIZAR USUARIO

```

CREATE TRIGGER UPDATE_USUARIO
AFTER UPDATE ON GSEG_USUARIO
FOR EACH ROW

```

```

BEGIN
    IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
            USU_CI = \', COALESCE(NEW.USU_CI, ''), '\ WHERE USU_CI = \', OLD.USU_CI
            , '\'), 'GSEG_USUARIO');
        END IF;

    IF(COALESCE(OLD.PAP_CODIGO, 0) <>
        COALESCE(NEW.PAP_CODIGO, 0)) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
            PAP_CODIGO = ', COALESCE(NEW.PAP_CODIGO, 0), ' WHERE USU_CI = \',
            OLD.USU_CI , '\'), 'GSEG_USUARIO');
        END IF;

    IF(COALESCE(OLD.USU_NOMBRE_USUARIO, "") <>
        COALESCE(NEW.USU_NOMBRE_USUARIO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
            USU_NOMBRE_USUARIO = \', COALESCE(NEW.USU_NOMBRE_USUARIO, ''),
            '\ WHERE USU_CI = \', OLD.USU_CI , '\'), 'GSEG_USUARIO');
        END IF;

    IF(COALESCE(OLD.USU_CLAVE, "") <> COALESCE(NEW.USU_CLAVE,
        "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
            USU_CLAVE = \', COALESCE(NEW.USU_CLAVE, ''), '\ WHERE USU_CI = \',
            OLD.USU_CI , '\'), 'GSEG_USUARIO');
        END IF;

    IF(COALESCE(OLD.USU_TIEMPO_CITA, 0) <>
        COALESCE(NEW.USU_TIEMPO_CITA, 0)) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
            USU_TIEMPO_CITA = ', COALESCE(NEW.USU_TIEMPO_CITA, 0), ' WHERE
            USU_CI = \', OLD.USU_CI , '\'), 'GSEG_USUARIO');
        END IF;

    IF(COALESCE(OLD.USU_ESTADO, "") <>
        COALESCE(NEW.USU_ESTADO, "")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
            RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
            USU_ESTADO = \', COALESCE(NEW.USU_ESTADO, ''), '\ WHERE USU_CI = \',
            OLD.USU_CI , '\'), 'GSEG_USUARIO');
        END IF;

    IF(COALESCE(OLD.USU_ACCION, "") <>
        COALESCE(NEW.USU_ACCION, "")) THEN

```

```

INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_ACCION = \'', COALESCE(NEW.USU_ACCION, ''), '\ ' WHERE USU_CI = \'',
OLD.USU_CI, '\ ' ), 'GSEG_USUARIO');
END IF;

```

```

IF(COALESCE(OLD.USU_NOMBRE1, '') <>
COALESCE(NEW.USU_NOMBRE1, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_NOMBRE1 = \'', COALESCE(NEW.USU_NOMBRE1, ''), '\ ' WHERE USU_CI
= \'', OLD.USU_CI, '\ ' ), 'GSEG_USUARIO');
END IF;

```

```

IF(COALESCE(OLD.USU_NOMBRE2, '') <>
COALESCE(NEW.USU_NOMBRE2, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_NOMBRE2 = \'', COALESCE(NEW.USU_NOMBRE2, ''), '\ ' WHERE USU_CI
= \'', OLD.USU_CI, '\ ' ), 'GSEG_USUARIO');
END IF;

```

```

IF(COALESCE(OLD.USU_APELLIDO1, '') <>
COALESCE(NEW.USU_APELLIDO1, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_APELLIDO1 = \'', COALESCE(NEW.USU_APELLIDO1, ''), '\ ' WHERE
USU_CI = \'', OLD.USU_CI, '\ ' ), 'GSEG_USUARIO');
END IF;

```

```

IF(COALESCE(OLD.USU_APELLIDO2, '') <>
COALESCE(NEW.USU_APELLIDO2, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_APELLIDO2 = \'', COALESCE(NEW.USU_APELLIDO2, ''), '\ ' WHERE
USU_CI = \'', OLD.USU_CI, '\ ' ), 'GSEG_USUARIO');
END IF;

```

```

IF(COALESCE(OLD.USU_EMAIL, '') <> COALESCE(NEW.USU_EMAIL,
'')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_EMAIL = \'', COALESCE(NEW.USU_EMAIL, ''), '\ ' WHERE USU_CI = \'',
OLD.USU_CI, '\ ' ), 'GSEG_USUARIO');
END IF;

```

```

IF(COALESCE(OLD.USU_CREATED_AT, '') <>
COALESCE(NEW.USU_CREATED_AT, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET

```

```

USU_CREATED_AT = \", COALESCE(NEW.USU_CREATED_AT, \"), '\ WHERE
USU_CI = \", OLD.USU_CI, \" ), 'GSEG_USUARIO');
    END IF;

```

```

    IF(COALESCE(OLD.USU_UPDATE_AT, \" ) <>
COALESCE(NEW.USU_UPDATE_AT, \")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE GSEG_USUARIO SET
USU_UPDATE_AT = \", COALESCE(NEW.USU_UPDATE_AT, \"), '\ WHERE
USU_CI = \", OLD.USU_CI, \" ), 'GSEG_USUARIO');
    END IF;

```

```

END
$$

```

INSERTAR INTERFERENCIA

```

CREATE TRIGGER INSERT_INTERFERENCIA
AFTER INSERT ON TPA_INTERFERENCIA
FOR EACH ROW
BEGIN
    DECLARE next_id INT;
    SET next_id = (SELECT AUTO_INCREMENT
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'testatencion'
AND TABLE_NAME = 'TPA_INTERFERENCIA') - 1;
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO
TPA_INTERFERENCIA (INT_ID, INT_NOMBRE, INT_DESCRIPCION, INT_TIPO,
INT_UBICACION, INT_ESTADO, INT_ACCION, INT_CREATED_AT,
INT_UPDATE_AT, USU_CI) VALUES (', next_id, ', \',
COALESCE(NEW.INT_NOMBRE, \"), '\, \',
COALESCE(NEW.INT_DESCRIPCION, \"), '\, \', COALESCE(NEW.INT_TIPO, \"),
'\, \', COALESCE(NEW.INT_UBICACION, \"), '\, \',
COALESCE(NEW.INT_ESTADO, \"), '\, \', COALESCE(NEW.INT_ACCION, \"), '\,
\', COALESCE(NEW.INT_CREATED_AT, \"), '\, \',
COALESCE(NEW.INT_UPDATE_AT, \"), '\, \', COALESCE(NEW.USU_CI, \"), '\')',
'TPA_INTERFERENCIA' );
END
$$

```

ACTUALIZAR INTERFERENCIA

```

CREATE TRIGGER UPDATE_INTERFERENCIA
AFTER UPDATE ON TPA_INTERFERENCIA
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.INT_ID, 0) <> COALESCE(NEW.INT_ID, 0)) THEN

```

```

INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_ID = ', COALESCE(NEW.INT_ID, 0), ' WHERE INT_ID = ', OLD.INT_ID),
'TPA_INTERFERENCIA');
END IF;

```

```

IF(COALESCE(OLD.INT_NOMBRE, "") <>
COALESCE(NEW.INT_NOMBRE, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_NOMBRE = \'', COALESCE(NEW.INT_NOMBRE, ""), '\ ' WHERE INT_ID
= ', OLD.INT_ID), 'TPA_INTERFERENCIA');
END IF;

```

```

IF(COALESCE(OLD.INT_DESCRIPCION, "") <>
COALESCE(NEW.INT_DESCRIPCION, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_DESCRIPCION = \'', COALESCE(NEW.INT_DESCRIPCION, ""), '\ '
WHERE INT_ID = ', OLD.INT_ID), 'TPA_INTERFERENCIA');
END IF;

```

```

IF(COALESCE(OLD.INT_TIPO, "") <> COALESCE(NEW.INT_TIPO, ""))
THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_TIPO = \'', COALESCE(NEW.INT_TIPO, ""), '\ ' WHERE INT_ID = ',
OLD.INT_ID), 'TPA_INTERFERENCIA');
END IF;

```

```

IF(COALESCE(OLD.INT_UBICACION, "") <>
COALESCE(NEW.INT_UBICACION, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_UBICACION = \'', COALESCE(NEW.INT_UBICACION, ""), '\ ' WHERE
INT_ID = ', OLD.INT_ID), 'TPA_INTERFERENCIA');
END IF;

```

```

IF(COALESCE(OLD.INT_ESTADO, "") <> COALESCE(NEW.INT_ESTADO,
"")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_ESTADO = \'', COALESCE(NEW.INT_ESTADO, ""), '\ ' WHERE INT_ID =
', OLD.INT_ID), 'TPA_INTERFERENCIA');
END IF;

```

```

IF(COALESCE(OLD.INT_ACCION, "") <> COALESCE(NEW.INT_ACCION,
"")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA

```

```

SET INT_ACCION = \", COALESCE(NEW.INT_ACCION, \"), \\' WHERE INT_ID = ',
OLD.INT_ID), 'TPA_INTERFERENCIA');
    END IF;

    IF(COALESCE(OLD.INT_CREATED_AT, \") <>
COALESCE(NEW.INT_CREATED_AT, \")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_CREATED_AT = \", COALESCE(NEW.INT_CREATED_AT, \"), \\'
WHERE INT_ID = ', OLD.INT_ID), 'TPA_INTERFERENCIA');
    END IF;

    IF(COALESCE(OLD.INT_UPDATE_AT, \") <>
COALESCE(NEW.INT_UPDATE_AT, \")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET INT_UPDATE_AT = \", COALESCE(NEW.INT_UPDATE_AT, \"), \\' WHERE
INT_ID = ', OLD.INT_ID), 'TPA_INTERFERENCIA');
    END IF;

    IF(COALESCE(OLD.USU_CI, \") <> COALESCE(NEW.USU_CI, \")) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_INTERFERENCIA
SET USU_CI = \", COALESCE(NEW.USU_CI, \"), \\' WHERE INT_ID = ',
OLD.INT_ID), 'TPA_INTERFERENCIA');
    END IF;

END
$$

```

INSERTAR INTERFERENCIA DETALLE

```

CREATE TRIGGER INSERT_INTERFERENCIA_DET
AFTER INSERT ON TPA_INTERFERENCIA_DET
FOR EACH ROW
BEGIN
    DECLARE next_id INT;
    SET next_id = (SELECT AUTO_INCREMENT
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'testatencion'
AND TABLE_NAME = 'TPA_INTERFERENCIA_DET') - 1;
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO
TPA_INTERFERENCIA_DET (DINT_ID, INT_ID, DINT_ORDEN,
DINT_DESCRIPCION, DINT_CORRECTA, DINT_UBICACION, DINT_ESTADO,
DINT_ACCION, DINT_CREATED_AT, DINT_UPDATE_AT, USU_CI) VALUES (',
next_id, ', ', COALESCE(NEW.INT_ID, 0), ', ', COALESCE(NEW.DINT_ORDEN, 0),
', \', COALESCE(NEW.DINT_DESCRIPCION, \"), \', \',
COALESCE(NEW.DINT_CORRECTA, \"), \', \',

```

```

COALESCE(NEW.DINT_UBICACION, ''), '\, \'
COALESCE(NEW.DINT_ESTADO, ''), '\, \'
COALESCE(NEW.DINT_ACCION, ''), '\, \'
COALESCE(NEW.DINT_CREATED_AT, ''), '\, \'
COALESCE(NEW.DINT_UPDATE_AT, ''), '\, \'
COALESCE(NEW.USU_CI, ''), '\ '
), 'TPA_INTERFERENCIA_DET' );
END
$$

```

ACTUALIZAR INTERFERENCIA DETALLE

```

CREATE TRIGGER UPDATE_INTERFERENCIA_DET
AFTER UPDATE ON TPA_INTERFERENCIA_DET
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.DINT_ID, 0) <> COALESCE(NEW.DINT_ID, 0))
    THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_ID = ', COALESCE(NEW.DINT_ID, 0), '
WHERE DINT_ID = ', OLD.DINT_ID ), 'TPA_INTERFERENCIA_DET');
        END IF;

        IF(COALESCE(OLD.INT_ID, 0) <> COALESCE(NEW.INT_ID, 0)) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET INT_ID = ', COALESCE(NEW.INT_ID, 0), '
WHERE DINT_ID = ', OLD.DINT_ID ), 'TPA_INTERFERENCIA_DET');
            END IF;

            IF(COALESCE(OLD.DINT_ORDEN, 0) <>
COALESCE(NEW.DINT_ORDEN, 0)) THEN
                INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_ORDEN = ',
COALESCE(NEW.DINT_ORDEN, 0), ' WHERE DINT_ID = ', OLD.DINT_ID ),
'TPA_INTERFERENCIA_DET');
                END IF;

                IF(COALESCE(OLD.DINT_DESCRIPCION, '') <>
COALESCE(NEW.DINT_DESCRIPCION, '')) THEN
                    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_DESCRIPCION = \'
COALESCE(NEW.DINT_DESCRIPCION, ''), '\ ' WHERE DINT_ID = ',
OLD.DINT_ID ), 'TPA_INTERFERENCIA_DET');
                    END IF;

                    IF(COALESCE(OLD.DINT_CORRECTA, '') <>
COALESCE(NEW.DINT_CORRECTA, '')) THEN

```

```

INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_CORRECTA = \'",
COALESCE(NEW.DINT_CORRECTA, ""), '\ WHERE DINT_ID = ', OLD.DINT_ID ),
'TPA_INTERFERENCIA_DET');
END IF;

```

```

IF(COALESCE(OLD.DINT_UBICACION, "") <>
COALESCE(NEW.DINT_UBICACION, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_UBICACION = \'",
COALESCE(NEW.DINT_UBICACION, ""), '\ WHERE DINT_ID = ', OLD.DINT_ID
), 'TPA_INTERFERENCIA_DET');
END IF;

```

```

IF(COALESCE(OLD.DINT_ESTADO, "") <>
COALESCE(NEW.DINT_ESTADO, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_ESTADO = \'",
COALESCE(NEW.DINT_ESTADO, ""), '\ WHERE DINT_ID = ', OLD.DINT_ID ),
'TPA_INTERFERENCIA_DET');
END IF;

```

```

IF(COALESCE(OLD.DINT_ACCION, "") <>
COALESCE(NEW.DINT_ACCION, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_ACCION = \'",
COALESCE(NEW.DINT_ACCION, ""), '\ WHERE DINT_ID = ', OLD.DINT_ID ),
'TPA_INTERFERENCIA_DET');
END IF;

```

```

IF(COALESCE(OLD.DINT_CREATED_AT, "") <>
COALESCE(NEW.DINT_CREATED_AT, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_CREATED_AT = \'",
COALESCE(NEW.DINT_CREATED_AT, ""), '\ WHERE DINT_ID = ',
OLD.DINT_ID ), 'TPA_INTERFERENCIA_DET');
END IF;

```

```

IF(COALESCE(OLD.DINT_UPDATE_AT, "") <>
COALESCE(NEW.DINT_UPDATE_AT, "")) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET DINT_UPDATE_AT = \'",
COALESCE(NEW.DINT_UPDATE_AT, ""), '\ WHERE DINT_ID = ', OLD.DINT_ID
), 'TPA_INTERFERENCIA_DET');

```

```

END IF;

IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE
TPA_INTERFERENCIA_DET SET USU_CI = \', COALESCE(NEW.USU_CI, ''), '\
WHERE DINT_ID = ', OLD.DINT_ID ), 'TPA_INTERFERENCIA_DET');
END IF;

END
$$

```

INSERTAR PREGUNTA

```

CREATE TRIGGER INSERT_PREGUNTA
AFTER INSERT ON TPA_PREGUNTA
FOR EACH ROW
BEGIN
    DECLARE next_id INT;
    SET next_id = (SELECT AUTO_INCREMENT
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'testatencion'
AND TABLE_NAME = 'TPA_PREGUNTA') - 1;
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO TPA_PREGUNTA
(PREG_ID, TEST_ID, PREG_ORDEN, PREG_EXPRESION, PREG_ESTADO,
PREG_ACCION, PREG_CREATED_AT, PREG_UPDATE_AT, USU_CI) VALUES
(', next_id, ', ', COALESCE(NEW.TEST_ID, 0), ', ',
COALESCE(NEW.PREG_ORDEN, 0), ', \', COALESCE(NEW.PREG_EXPRESION,
'), '\, \', COALESCE(NEW.PREG_ESTADO, ''), '\, \',
COALESCE(NEW.PREG_ACCION, ''), '\, \',
COALESCE(NEW.PREG_CREATED_AT, ''), '\, \',
COALESCE(NEW.PREG_UPDATE_AT, ''), '\, \', COALESCE(NEW.USU_CI, ''), '\
)'), 'TPA_PREGUNTA' );
END
$$

```

ACTUALIZAR PREGUNTA

```

CREATE TRIGGER UPDATE_PREGUNTA
AFTER UPDATE ON TPA_PREGUNTA
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.PREG_ID, 0) <> COALESCE(NEW.PREG_ID, 0))
THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET

```

```

PREG_ID = ', COALESCE(NEW.PREG_ID, 0), ' WHERE PREG_ID = ',
OLD.PREG_ID ), 'TPA_PREGUNTA');
    END IF;

    IF(COALESCE(OLD.TEST_ID, 0) <> COALESCE(NEW.TEST_ID, 0))
THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
TEST_ID = ', COALESCE(NEW.TEST_ID, 0), ' WHERE PREG_ID = ',
OLD.PREG_ID ), 'TPA_PREGUNTA');
    END IF;

    IF(COALESCE(OLD.PREG_ORDEN, 0) <>
COALESCE(NEW.PREG_ORDEN, 0)) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
PREG_ORDEN = ', COALESCE(NEW.PREG_ORDEN, 0), ' WHERE PREG_ID = ',
OLD.PREG_ID ), 'TPA_PREGUNTA');
    END IF;

    IF(COALESCE(OLD.PREG_EXPRESION, "") <>
COALESCE(NEW.PREG_EXPRESION, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
PREG_EXPRESION = \'", COALESCE(NEW.PREG_EXPRESION, ""), \' WHERE
PREG_ID = ', OLD.PREG_ID ), 'TPA_PREGUNTA');
    END IF;

    IF(COALESCE(OLD.PREG_ESTADO, "") <>
COALESCE(NEW.PREG_ESTADO, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
PREG_ESTADO = \'", COALESCE(NEW.PREG_ESTADO, ""), \' WHERE PREG_ID
= ', OLD.PREG_ID ), 'TPA_PREGUNTA');
    END IF;

    IF(COALESCE(OLD.PREG_ACCION, "") <>
COALESCE(NEW.PREG_ACCION, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
PREG_ACCION = \'", COALESCE(NEW.PREG_ACCION, ""), \' WHERE PREG_ID =
', OLD.PREG_ID ), 'TPA_PREGUNTA');
    END IF;

    IF(COALESCE(OLD.PREG_CREATED_AT, "") <>
COALESCE(NEW.PREG_CREATED_AT, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
PREG_CREATED_AT = \'", COALESCE(NEW.PREG_CREATED_AT, ""), \' WHERE
PREG_ID = ', OLD.PREG_ID ), 'TPA_PREGUNTA');

```

```

END IF;

IF(COALESCE(OLD.PREG_UPDATE_AT, "") <>
COALESCE(NEW.PREG_UPDATE_AT, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
PREG_UPDATE_AT = \', COALESCE(NEW.PREG_UPDATE_AT, ''), '\ ' WHERE
PREG_ID = ', OLD.PREG_ID ), 'TPA_PREGUNTA');
END IF;

IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_PREGUNTA SET
USU_CI = \', COALESCE(NEW.USU_CI, ''), '\ ' WHERE PREG_ID = ',
OLD.PREG_ID ), 'TPA_PREGUNTA');
END IF;

END
$$

```

INSERTAR RESPUESTA

```

CREATE TRIGGER INSERT_RESPUESTA
AFTER INSERT ON TPA_RESPUESTA
FOR EACH ROW
BEGIN
    DECLARE next_id INT;
    SET next_id = (SELECT AUTO_INCREMENT
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_SCHEMA = 'testatencion'
AND TABLE_NAME = 'TPA_RESPUESTA') - 1;
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO TPA_RESPUESTA
(Resp_ID, TEST_ID, PREG_ID, Resp_ORDEN, Resp_UBICACION,
Resp_CORRECTA, Resp_ESTADO, Resp_ACCION, Resp_CREATED_AT,
Resp_UPDATE_AT, USU_CI) VALUES (', next_id, ', ',
COALESCE(NEW.TEST_ID, 0), ', ', COALESCE(NEW.PREG_ID, 0), ', ',
COALESCE(NEW.Resp_ORDEN, 0), ', \', COALESCE(NEW.Resp_UBICACION,
''), '\, \', COALESCE(NEW.Resp_CORRECTA, ''), '\, \',
COALESCE(NEW.Resp_ESTADO, ''), '\, \', COALESCE(NEW.Resp_ACCION, ''),
'\, \', COALESCE(NEW.Resp_CREATED_AT, ''), '\, \',
COALESCE(NEW.Resp_UPDATE_AT, ''), '\, \', COALESCE(NEW.USU_CI, ''), '\ '
), 'TPA_RESPUESTA' );
END
$$

```

ACTUALIZAR RESPUESTA

```
CREATE TRIGGER UPDATE_RESPUESTA
AFTER UPDATE ON TPA_RESPUESTA
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.RESP_ID, 0) <> COALESCE(NEW.RESP_ID, 0))
    THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_ID = ', COALESCE(NEW.RESP_ID, 0), ' WHERE RESP_ID = ',
OLD.RESP_ID ), 'TPA_RESPUESTA');
        END IF;

        IF(COALESCE(OLD.TEST_ID, 0) <> COALESCE(NEW.TEST_ID, 0))
    THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
TEST_ID = ', COALESCE(NEW.TEST_ID, 0), ' WHERE RESP_ID = ',
OLD.RESP_ID ), 'TPA_RESPUESTA');
        END IF;

        IF(COALESCE(OLD.PREG_ID, 0) <> COALESCE(NEW.PREG_ID, 0))
    THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
PREG_ID = ', COALESCE(NEW.PREG_ID, 0), ' WHERE RESP_ID = ',
OLD.RESP_ID ), 'TPA_RESPUESTA');
        END IF;

        IF(COALESCE(OLD.RESP_ORDEN, 0) <>
COALESCE(NEW.RESP_ORDEN, 0)) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_ORDEN = ', COALESCE(NEW.RESP_ORDEN, 0), ' WHERE RESP_ID = ',
OLD.RESP_ID ), 'TPA_RESPUESTA');
            END IF;

        IF(COALESCE(OLD.RESP_UBICACION, "") <>
COALESCE(NEW.RESP_UBICACION, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_UBICACION = \'', COALESCE(NEW.RESP_UBICACION, ""), '\ ' WHERE
RESP_ID = ', OLD.RESP_ID ), 'TPA_RESPUESTA');
            END IF;

        IF(COALESCE(OLD.RESP_CORRECTA, "") <>
COALESCE(NEW.RESP_CORRECTA, "")) THEN
            INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
```

```

RESP_CORRECTA = '\', COALESCE(NEW.RESP_CORRECTA, ''), '\ WHERE
RESP_ID = ', OLD.RESP_ID ), 'TPA_RESPUESTA');
    END IF;

    IF(COALESCE(OLD.RESP_ESTADO, '') <>
COALESCE(NEW.RESP_ESTADO, '')) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_ESTADO = '\', COALESCE(NEW.RESP_ESTADO, ''), '\ WHERE RESP_ID =
', OLD.RESP_ID ), 'TPA_RESPUESTA');
    END IF;

    IF(COALESCE(OLD.RESP_ACCION, '') <>
COALESCE(NEW.RESP_ACCION, '')) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_ACCION = '\', COALESCE(NEW.RESP_ACCION, ''), '\ WHERE RESP_ID =
', OLD.RESP_ID ), 'TPA_RESPUESTA');
    END IF;

    IF(COALESCE(OLD.RESP_CREATED_AT, '') <>
COALESCE(NEW.RESP_CREATED_AT, '')) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_CREATED_AT = '\', COALESCE(NEW.RESP_CREATED_AT, ''), '\ WHERE
RESP_ID = ', OLD.RESP_ID ), 'TPA_RESPUESTA');
    END IF;

    IF(COALESCE(OLD.RESP_UPDATE_AT, '') <>
COALESCE(NEW.RESP_UPDATE_AT, '')) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
RESP_UPDATE_AT = '\', COALESCE(NEW.RESP_UPDATE_AT, ''), '\ WHERE
RESP_ID = ', OLD.RESP_ID ), 'TPA_RESPUESTA');
    END IF;

    IF(COALESCE(OLD.USU_CI, '') <> COALESCE(NEW.USU_CI, '')) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_RESPUESTA SET
USU_CI = '\', COALESCE(NEW.USU_CI, ''), '\ WHERE RESP_ID = ',
OLD.RESP_ID ), 'TPA_RESPUESTA');
    END IF;

END
$$

```

INSERTAR TEST

```
CREATE TRIGGER INSERT_TEST
AFTER INSERT ON TPA_TEST
FOR EACH ROW
BEGIN
    DECLARE next_id INT;
    SET next_id = (SELECT AUTO_INCREMENT
                   FROM INFORMATION_SCHEMA.TABLES
                   WHERE TABLE_SCHEMA = 'testatencion'
                   AND TABLE_NAME = 'TPA_TEST') - 1;
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES ( CONCAT ('INSERT INTO TPA_TEST
(TEST_ID, TEST_NOMBRE, TEST_DESCRIPCION, TEST_NUM_FILAS,
TEST_INSTRUCCIONES, TEST_ESTADO, TEST_ACCION,
TEST_INTERFERENCIA, TEST_TIEMPO_INTER, TEST_CREATED_AT,
TEST_UPDATE_AT, USU_CI) VALUES (', next_id, ',\ ',
COALESCE(NEW.TEST_NOMBRE, ''), '\, \ ',
COALESCE(NEW.TEST_DESCRIPCION, ''), '\, \ ',
COALESCE(NEW.TEST_NUM_FILAS, 0), ', \ ',
COALESCE(NEW.TEST_INSTRUCCIONES, 0), '\, \ ',
COALESCE(NEW.TEST_ESTADO, ''), '\, \ ', COALESCE(NEW.TEST_ACCION, ''),
'\, \ ', COALESCE(NEW.TEST_INTERFERENCIA, ''), '\, \ ',
COALESCE(NEW.TEST_TIEMPO_INTER, ''), '\, \ ',
COALESCE(NEW.TEST_CREATED_AT, ''), '\, \ ',
COALESCE(NEW.TEST_UPDATE_AT, ''), '\, \ ', COALESCE(NEW.USU_CI, ''), '\ '
), 'TPA_TEST' );
END
$$
```

ACTUALIZAR TEST

```
CREATE TRIGGER UPDATE_TEST
AFTER UPDATE ON TPA_TEST
FOR EACH ROW
BEGIN
    IF(COALESCE(OLD.TEST_ID, 0) <> COALESCE(NEW.TEST_ID, 0))
    THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET TEST_ID
= ', COALESCE(NEW.TEST_ID, 0), ' WHERE TEST_ID = ', OLD.TEST_ID),
'TPA_TEST');
        END IF;

    IF(COALESCE(OLD.TEST_NOMBRE, '') <>
COALESCE(NEW.TEST_NOMBRE, '')) THEN
        INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
```

```

TEST_NOMBRE = \', COALESCE(NEW.TEST_NOMBRE, ''), '\ WHERE TEST_ID
= ', OLD.TEST_ID ), 'TPA_TEST');
END IF;

```

```

IF(COALESCE(OLD.TEST_DESCRIPCION, '') <>
COALESCE(NEW.TEST_DESCRIPCION, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_DESCRIPCION = \', COALESCE(NEW.TEST_DESCRIPCION, ''), '\ WHERE
TEST_ID = ', OLD.TEST_ID ), 'TPA_TEST');
END IF;

```

```

IF(COALESCE(OLD.TEST_NUM_FILAS, 0) <>
COALESCE(NEW.TEST_NUM_FILAS, 0)) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_NUM_FILAS = ', COALESCE(NEW.TEST_NUM_FILAS, 0), ' WHERE
TEST_ID = ', OLD.TEST_ID), 'TPA_TEST');
END IF;

```

```

IF(COALESCE(OLD.TEST_INSTRUCCIONES, 0) <>
COALESCE(NEW.TEST_INSTRUCCIONES, 0)) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_INSTRUCCIONES = ', COALESCE(NEW.TEST_INSTRUCCIONES, 0), '
WHERE TEST_ID = ', OLD.TEST_ID), 'TPA_TEST');
END IF;

```

```

IF(COALESCE(OLD.TEST_ESTADO, '') <>
COALESCE(NEW.TEST_ESTADO, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_ESTADO = \', COALESCE(NEW.TEST_ESTADO, ''), '\ WHERE TEST_ID =
', OLD.TEST_ID ), 'TPA_TEST');
END IF;

```

```

IF(COALESCE(OLD.TEST_ACCION, '') <>
COALESCE(NEW.TEST_ACCION, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_ACCION = \', COALESCE(NEW.TEST_ACCION, ''), '\ WHERE TEST_ID =
', OLD.TEST_ID ), 'TPA_TEST');
END IF;

```

```

IF(COALESCE(OLD.TEST_INTERFERENCIA, '') <>
COALESCE(NEW.TEST_INTERFERENCIA, '')) THEN
INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_INTERFERENCIA = \', COALESCE(NEW.TEST_INTERFERENCIA, ''), '\
WHERE TEST_ID = ', OLD.TEST_ID ), 'TPA_TEST');

```

```

END IF;

IF(COALESCE(OLD.TEST_TIEMPO_INTER, 0) <>
COALESCE(NEW.TEST_TIEMPO_INTER, 0)) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_TIEMPO_INTER = ', COALESCE(NEW.TEST_TIEMPO_INTER, 0), '
WHERE TEST_ID = ', OLD.TEST_ID), 'TPA_TEST');
    END IF;

IF(COALESCE(OLD.TEST_CREATED_AT, "") <>
COALESCE(NEW.TEST_CREATED_AT, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_CREATED_AT = \'', COALESCE(NEW.TEST_CREATED_AT, ""), '\ ' WHERE
TEST_ID = ', OLD.TEST_ID ), 'TPA_TEST');
    END IF;

IF(COALESCE(OLD.TEST_UPDATE_AT, "") <>
COALESCE(NEW.TEST_UPDATE_AT, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET
TEST_UPDATE_AT = \'', COALESCE(NEW.TEST_UPDATE_AT, ""), '\ ' WHERE
TEST_ID = ', OLD.TEST_ID ), 'TPA_TEST');
    END IF;

IF(COALESCE(OLD.USU_CI, "") <> COALESCE(NEW.USU_CI, "")) THEN
    INSERT INTO GSEG_REG_MOVIMIENTOS(RMOV_SENTENCIA,
RMOV_NOM_TABLA) VALUES (CONCAT ('UPDATE TPA_TEST SET USU_CI =
\'', COALESCE(NEW.USU_CI, ""), '\ ' WHERE TEST_ID = ', OLD.TEST_ID ),
'TPA_TEST');
    END IF;

END
$$

```

Fotos de las pruebas hechas con estudiantes del Colegio Latinoamericano



