**INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA ZACATECAS

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**SiCMA**

**Sistema para el cálculo de medidas antropométricas basado en ISAK 2**

Plan de Pruebas

PRESENTA:

**Montserrat Silva Cordero**

**Hilario Abraham Rodarte España**

**XX de XX del 2020**

Índice

[Control de cambios 3](#_Toc55440783)

[Introducción 3](#_Toc55440784)

[Propósito 3](#_Toc55440785)

[Alcance 3](#_Toc55440786)

[Referencias 3](#_Toc55440787)

[Estructuración de pruebas 4](#_Toc55440788)

[Definición de pruebas 5](#_Toc55440789)

[Identificación de elementos para ambiente de pruebas 7](#_Toc55440790)

[Computadores 7](#_Toc55440791)

[Dispositivos móbiles 8](#_Toc55440792)

[Software 9](#_Toc55440793)

[Otros 10](#_Toc55440794)

[Pruebas Unitarias 12](#_Toc55440795)

[Pruebas Integración 34](#_Toc55440796)

[Pruebas de Sistema 34](#_Toc55440797)

Control de cambios

Registro del control de cambios en el documento SRS del Sistema para el cálculo de medidas antropométricas basado en ISAK 2 (SiCMA)

|  |  |  |  |
| --- | --- | --- | --- |
| No. Revisión | Descripción | Fecha | Estatus |
| 01 | Versión inicial previa a la revisión del E.T, planteamiento de pruebas. | Xx/xx/xx | Aprobado |
| 02 | Versión posterior a revisión con integración de definición de pruebas. | Xx/xx/xx | Pendiente revisión |

Introducción

Propósito

Alcance

Referencias

[1] M. Silva Cordero y H. A. Rodarte España, “Diseño de Software para el Sistema para el Cálculo de Medidadas Antropométricas Basado en ISAK 2 ‘SICMA’”, 2020.

[2] M. Silva Cordero y H. A. Rodarte España, “Plan de Pruebas del Sistema para el Cálculo de Medidas Antropométricas basado en ISAK 2”, 2020.

[3] M. Silva Cordero y H. A. Rodarte España, “Diccionario de datos para el Sistema para el cálculo de medidas antropométricas basado en ISAK 2”, 2020.

[4] Silva Cordero, Montserrat y H. A. Rodarte España, “Pruebas Unitarias para el Sistema para el cálculo de medidas antropométricas basado en ISAK 2”, 2020.

Estructuración de pruebas

En esta sección será presentada la estructura diseñada para la delimitación y aplicación de pruebas unitarias, de integración y de sistema en SICMA.

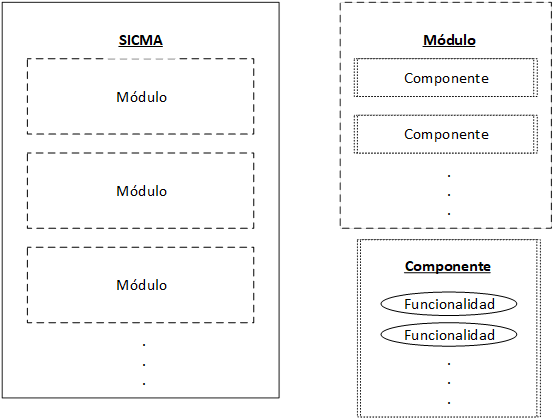


Ilustración 1 Esquema de estructuración de módulos.

El sistema SICMA está conformado por módulos, los cuales son definidos en el “Diagrama de Componentes” en [1], derivado de esta modularidad las pruebas fueron diseñadas para la comprobación del funcionamiento de cada uno de estos módulos. Tal y como se muestra en Ilustración 2 cada módulo contiene una serie de componentes, de los cuales derivan distintas funcionalidades.

Comprendiendo el objetivo de cada uno de los tipos de pruebas a realizar se estipulo la aplicación de pruebas unitarias a cada funcionalidad, pruebas de integración por cada componente y las pruebas de sistema por cada módulo, además de al funcionamiento de estos módulos integrados, como ejemplo grafico de esta distribución se muestra la Ilustración 2.

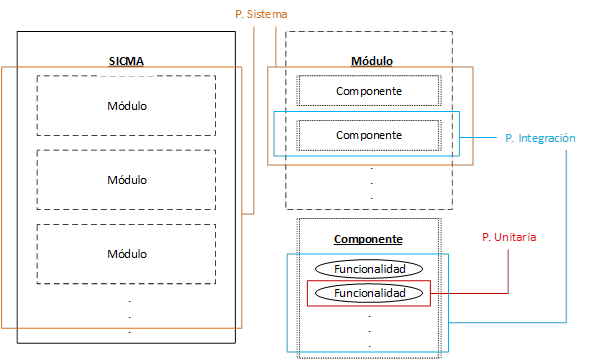


Ilustración 2 Esquema de aplicación de pruebas

Definición de pruebas

En esta sección serán presentadas las descripciones y procedimientos de pruebas diseñados para el sistema.

Para cada prueba fue diseñado un reporte de especificación y resultados presentados en [2], el cual deberá ser utilizado y llenado al momento de ejecutarse dichas pruebas, a continuación en Tabla 1 se muestra una breve explicación de la estructura manejada en dicho reporte.

Tabla 1 Definición de sección "Especificación de Prueba"

|  |  |
| --- | --- |
| Módulo: | Identificador del módulo al que pertenece la prueba |
| Componente: | Identificador del componente al que pertenece la prueba. |
| Id: | Identificador de la prueba. |
| Fecha: | Fecha en la que se diseñó o rediseño la prueba  *\*Es necesario mantener registrada cada fecha de cambio.* |
| Nombre: | Nombre de identificación de la prueba. |
| Autor: | Autor del diseño de la prueba. |
| Requerimiento para atender: | Se enlistan los Identificadores de requerimientos atendidos con la prueba. |
| Tipo de prueba: | Se indica el tipo de prueba que es.  <Prueba unitaria.>  <Prueba de integración>  <Prueba de sistema> |
| Técnica de prueba: | Se indica la técnica de la prueba.  <Caja negra>  <Caja blanca> |
| Objetivo: | Objetivo de realización de la prueba. |
| Ambiente de pruebas: | Se enlistan los identificadores de los elementos requeridos para realizar la prueba. |
| Criterio de aceptación: | Se establece la condición de aceptación de la prueba como finalizada. |
| Criterio de inicio o reanudación: | Se establece la condición de inicio de la prueba o de reanudación según sea el caso. |
| Criterio de suspensión: | Se establece la condición de suspensión de la aplicación de la prueba. |
| Estado: | [Estado]: Se indica el estado en el que se encuentra la prueba.  <No realizada>: Aun no es aplicada la prueba  <En prueba>: Se encuentra realizando los casos de prueba.  <Suspendida>: Se comenzó la realización, pero fue interrumpida.  <Finalizada>: Se concluyo la aplicación de la prueba.  [Fecha]: Fecha en la que se ejecutó el cambio de estado.  *\*Es necesario mantener registrada cada fecha y estado de cambio.* |
| Resultado: | Se indica el resultado final de la prueba  <Sin definir>: No se tienen resultados  <Exitoso>: La prueba cumplió con los criterios de aceptación.  <No Exitoso>: La prueba no cumplió con los criterios de aceptación. |
| Errores asociados: | Al ejecutar el caso de prueba, en este campo se indican los identificadores únicos de los errores presentados. |

*Fuente: Elaboración propia.*

Identificación de elementos para ambiente de pruebas

Nomenclatura. - Los recursos de hardware y software que se utilizarán para la realización de pruebas son enlistados a continuación, bajo el etiquetado R0X.

Computadores

• **R01.- Notebook HP**

o Procesador: I5-6200U 2.3GHz x 4

o Sistema Operativo: Ubuntu 16.04 LTS + Windows 10

o RAM:8GB

o Almacenamiento:1TB

• **R02.- Notebook DELL**

o Procesador: I5-7200U 2.50GHz x4

o Sistema Operativo: Ubuntu 16.04 LTS + Windows 10

o RAM: 8GB

o Almacenamiento: 1TB

• **R03.- Computadora de Escritorio**

o Procesador: Ryzen 5 1500X 3.5GHz x 4

o Sistema Operativo: Ubuntu 16.04 LTS + Windows 10

o RAM:8GB

o Almacenamiento:840GB

Dispositivos móbiles

• **R04.- Smartphone Redmi note 7**

o Procesador: Octa-core (4x2.2 GHz Kryo 260 & 4x1.8 GHz Kryo 260)

o Sistema Operativo: Android 9

o RAM:4GB

o Almacenamiento:64GB

o Cámara trasera: Dual 48 MP (f/1.8)+ 5 MP (f/2.4)

o Bluetooth: 5.0

• **R05.- Smartphone Samsung Galaxy A51**

o Procesador: Octa-core (2.3Ghz Quad-Core ARM Cortex-A73 & 1.7Ghz Quad-Core ARM Cortex-A53)

o Sistema Operativo: Android 10

o RAM:4GB

o Almacenamiento:128GB

o Cámara trasera: (quad cámara) 48 MP + 12MP ultrawide (f/2.2 13mm 1.12µm)+5MP telephoto (f/2.4 40mm)+5MP depth sensor (f/2.2 1/5.0" 1.12µm)

o Bluetooth:5.0 con A2DP/LE

• **R12.- Smartphone Huawei p30**

o Procesador: Kirin 710 (4x2.2GHz + 4x1.7GHz).

o Sistema Operativo: Android 10

o RAM:4GB

o Almacenamiento:128GB

o Cámara trasera: 24 MP (f/1.8) + gran angular de 8 MP (f/2.4) + detección de profundidad de 8 MP (f/2.4).

o Bluetooth: 4.2

• **R13.- Smartphone Moto Z**

o Procesador: Snapdragon 820 Quad Core (2x2.15GHz + 2x1.6GHz)

o Sistema Operativo: Android 8

o RAM:4GB

o Almacenamiento:32GB

o Cámara trasera: 13 MP

o Bluetooth: 4.1

Software

• **R8.- Paquete de ofimática: Microsoft Office 365**

o Versión: actualización más reciente.

• **R9.- IDE de desarrollo: Visual Studio Code**

o Versión: 1.46 o actualización más reciente.

• **R10.- Software de Arduino**

o Versión: 1.8.13 o actualización más reciente.

• **R14.- GitHub**

o Sistema de gestión de proyectos y control de versiones de código.

• **R15.- SICMA (código fuente)**

* Versión: 0.01 con actualización más reciente.

• **R15.01.-Base de datos de SICMA**

* Versión: 0.01 con actualización más reciente.

• **R16.- Herramienta para generación de datos. (código fuente)**

* Versión: 0.01 con actualización más reciente.

Otros

• **R6.- Plicómetro**

o Alcance de medición: 0 -88 mm

o Dimensiones: 27cm x 25cm x 1cm

o Peso: 150 grs.

• **R7.- Placa Arduino**

o Microcontrolador: ATmega328P – 8 bit AVR family microcontroller

o Voltaje :5V

o Voltaje recomendado para los pines: 7-12V

o Pines análogos de entrada: 6

o Pines digitales de entrada y salida: 14

o Corriente en pines de entrada y salida: 40 mA

o Frecuencia (Velocidad de reloj):16 MHz

• **R16.- Módulo bluetooth**

• **R17.- Potenciómetro de precisión**

Pruebas Unitarias

**Índice de Pruebas**

[Identificación de elementos 15](#_Toc55436182)

[**MD01.01.- Aplicación móvil** 16](#_Toc55436183)

[**CM01.01.- Sistema de visión artificial** 16](#_Toc55436184)

[*PU01.- Comunicación con la cámara del dispositivo* 16](#_Toc55436185)

[*PU02.- Identificación de puntos de referencia* 16](#_Toc55436186)

[*PU03.- Toma de fotografía* 16](#_Toc55436187)

[**CM02.01.- Sistema de reconocimiento del cuerpo humano** 16](#_Toc55436188)

[*PU04.- Lectura de fotografía* 16](#_Toc55436189)

[*PU05.- Preprocesamiento de la imagen* 16](#_Toc55436190)

[*PU06.- Procesamiento de la imagen* 16](#_Toc55436191)

[*PU07.- Cálculo de medidas* 16](#_Toc55436192)

[*PU39.- Implementación de fórmulas* 16](#_Toc55436193)

[*PU40.- Clasificación de resultados por índices* 16](#_Toc55436194)

[**CM03.01.- Interfaz gráfica de usuario** 16](#_Toc55436195)

[*PU08.- Secuencia y enrutamiento de pantallas* 16](#_Toc55436196)

[*PU09.- Repuesta de formulario* 17](#_Toc55436197)

[*PU10.- Respuesta de teclado* 17](#_Toc55436198)

[*PU12.- Persistencia de datos* 17](#_Toc55436199)

[**CM04.01.- Modulo bluetooth** 18](#_Toc55436200)

[*PU13.- Comunicación con modulo bluetooth del dispositivo.* 19](#_Toc55436201)

[*PU14.- Recepción de datos.* 19](#_Toc55436202)

[**CM05.01.- Base de datos** 19](#_Toc55436203)

[*PU15.- Creación de base de datos local en dispositivo* 19](#_Toc55436204)

[*PU16.- Comunicación aplicación-base de datos* 19](#_Toc55436205)

[*PU17.- Altas de Pacientes* 21](#_Toc55436206)

[*PU18.- Altas de Historial Clínico* 24](#_Toc55436207)

[*PU19.- Altas de Consulta* 28](#_Toc55436208)

[***Bajas de registros*** 30](#_Toc55436209)

[*PU20.-Bajas de Pacientes* 30](#_Toc55436210)

[***Cambios en registros*** 31](#_Toc55436211)

[*PU21.- Cambios de Pacientes* 31](#_Toc55436212)

[*PU22.- Cambios de Historial Clínico* 32](#_Toc55436213)

[*PU23.- Cambios de Consultas* 33](#_Toc55436214)

[***Consultas en registros*** 34](#_Toc55436215)

[*PU24.- Consulta de Pacientes* 34](#_Toc55436216)

[*PU25.- Consulta de Historial Clínico* 35](#_Toc55436217)

[*PU26.- Consulta de Consultas* 36](#_Toc55436218)

[*PU27.- Eliminación de base de datos local en dispositivo* 37](#_Toc55436219)

[**MD02.01.- Sistema embebido** 38](#_Toc55436220)

[**CM06.01.- Sensor angular** 38](#_Toc55436221)

[*PU28.-Funcionalidad* 38](#_Toc55436222)

[*PU29.- Precisión* 38](#_Toc55436223)

[**CM07.01.- Arduino** 38](#_Toc55436224)

[*PU30.- Funcionalidad* 38](#_Toc55436225)

[*PU31.- Comunicación con módulo bluetooth* 38](#_Toc55436226)

[*PU32.- Envío de datos con módulo bluetooth* 38](#_Toc55436227)

[*PU33.- Comunicación con sensor angular* 38](#_Toc55436228)

[**CM08.01.- Módulo bluetooth** 38](#_Toc55436229)

[*PU34.- Funcionalidad* 38](#_Toc55436230)

[*PU35.- Envío de datos.* 38](#_Toc55436231)

[**MD03.01.- Google Drive** 38](#_Toc55436232)

[**CM09.01.- Respaldo y restauración** 38](#_Toc55436233)

[*PU36.- Comunicación con el servicio* 38](#_Toc55436234)

[*PU37.- Respaldo de información* 38](#_Toc55436235)

[*PU38.- Recuperación de información* 38](#_Toc55436236)

**Identificación de elementos**

*Nomenclatura y versionamiento. -* Los módulos y componentes del proyecto están etiquetados como MD0X y CM0X respectivamente; y su versión se verá reflejada en la numeración posterior al punto (.) dentro de su nomenclatura. Cada componente presenta el identificador (PU0X) y nombre de las pruebas unitaria que lo componen.

**MD01.01.- Aplicación móvil**

**CM01.01.- Sistema de visión artificial**

Las pruebas del sistema de visión artificial, listadas a continuación están hechas para la validación y verificación del correcto funcionamiento de este módulo en específico.

*PU01.- Comunicación con la cámara del dispositivo*

Prueba para la validación de la comunicación de la cámara del dispositivo con la aplicación móvil.

*PU02.- Identificación de puntos de referencia*

Prueba para la validación de la obtención de los puntos de referencia de una imagen.

*PU03.- Toma de fotografía*

Prueba para la validación de la manipulación de la cámara del dispositivo, así como la obtención de la imagen.

**CM02.01.- Sistema de reconocimiento del cuerpo humano**

Las pruebas tomadas en cuenta para la validación y verificación del correcto funcionamiento del sistema de reconocimiento del cuerpo humano son listadas a continuación.

*PU04.- Lectura de fotografía*

Prueba para la validación de la correcta lectura de la fotografía tomando en cuenta la apertura para la lectura de información.

*PU05.- Preprocesamiento de la imagen*

Prueba para la validación del preprocesamiento de la imagen para sus futuros usos.

*PU06.- Procesamiento de la imagen*

Prueba para la validación del procesamiento de la imagen para la extracción de información.

*PU07.- Cálculo de medidas*

Prueba para la validación y obtención de los datos establecidos producido por el cálculo de medidas antropométricas.

*PU39.- Implementación de fórmulas*

*PU40.- Clasificación de resultados por índices*

**CM03.01.- Interfaz gráfica de usuario**

*PU08.- Secuencia y enrutamiento de pantallas*

***PU08.01.-Caja Blanca: Análisis estático de código.***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

***PU08.02.-Caja Blanca: Cobertura de Caminos.***

La prueba será realizada mediante la utilización de ***widget test***, proporcionadas por la herramienta de desarrollo, aquí se escriben todos los casos de prueba suficientes para que se ejecuten todos los caminos de secuencia de pantallas. La validación de esta prueba se generará a partir de la presentación de banderas detonadas al pasar por una pantalla.

*PU09.- Repuesta de interacción de formulario*

***PU09.01.-Caja Blanca***

La prueba será realizada mediante la utilización de ***widget test***, proporcionadas por la herramienta de desarrollo, en ellos se comprobarán los elementos de los formularios presentados en las pantallas de la aplicación. La validación de esta prueba se generará a partir de la presentación de banderas detonadas al ejecutarse correctamente cada elemento del formulario.

Los elementos de input considerados en los formularios son los siguientes:

* Texto
* Selectivo o de despliegue
* Radio button o de cambio de estado
* botón

Otros elementos a evaluar será:

* Etiqueta

Dentro de esta prueba se validará su correcta aparición, y funcionamiento, ya sea obtención de información (ej. Input text), accionador de funciones (ej. botón) o presentador de información (ej. etiqueta).

*PU10.- Respuesta de interacción de teclado*

***PU10.01.-Caja Negra***

La prueba será realizada mediante la utilización de ***widget test***, proporcionadas por la herramienta de desarrollo, en ellos se comprobarán los elementos de los formularios presentados en las pantallas de la aplicación. Consistirá en la simulación de inserción de data a cada input que requiera del teclado, con el fin de validar su correcta aparición y presentación.

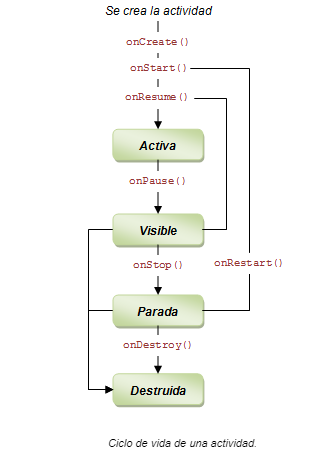
Dentro de esta prueba se validará su correcta aparición, y funcionamiento, ya sea obtención de información (ej. Input text), accionador de funciones (ej. botón) o presentador de información (ej. etiqueta).

*PU12.- Persistencia de datos*

***PU1.01.-Caja Negra***

La prueba será realizada mediante la utilización de ***widget test***, proporcionadas por la herramienta de desarrollo, en ellos se comprobarán los elementos de los formularios presentados en las pantallas de la aplicación. Consistirá en la simulación de inserción de data a cada input, posteriormente generando interrupciones de acuerdo con el ciclo de vida de la aplicación y los estados que se presenta, con el fin de validar la persistencia de datos estas interrupciones.

El ciclo de vida para una aplicación en Android se presenta de la siguiente manera:



Se implementará bajo este esquema la interrupción de la ejecución de la aplicación mediante el respectivo clico de Flutter.

**CM04.01.- Modulo bluetooth**

Se establecen las pruebas pensadas para la validación y el correcto funcionamiento del módulo bluetooth.

*PU13.- Comunicación con modulo bluetooth del dispositivo.*

Prueba establecida para validar la correcta manipulación del módulo bluetooth del dispositivo móvil para la recepción de datos.

*PU14.- Recepción de datos.*

Prueba para la validación de la integridad de los datos recibidos por el módulo bluetooth.

**CM05.01.- Base de datos**

*PU15.- Creación de base de datos local en dispositivo*

***PU15.01.- Caja Negra:*** ***basada en fallas***

La prueba desarrollada para la comprobación de la creación correcta de la base de datos se hará uso de las herramientas proporcionadas por Flutter para la realización de pruebas, se desarrollará un ***unit test*** para ser ejecutado a la clase “*DBController”* específicamente a la función *“crearDB”*, donde se buscará validar la correcta funcionalidad de esta.

Los errores evaluados para esta prueba son:

* ER01.-No existencia de nombre de base de datos.
* ER02.-No existencia de esquema.
* ER03.-No existencia de usuario de base de datos.
* ER04.-No existencia de contraseña de base de datos.
* ER05.-No duplicidad de bases de datos.
* ER06.-No creación de base de datos

Cada caso de evaluación se encuentran descritos en [2], cada tester deberá ejecutar el ***unit test*** programado y presentar resultado en el documento.

***PU15.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU16.- Comunicación aplicación-base de datos*

***PU16.01.- Caja Negra:*** ***basada en fallas***

La prueba desarrollada para la comprobación de la creación correcta de la base de datos se hará uso de las herramientas proporcionadas por Flutter para la realización de pruebas, se desarrollará un ***unit test*** para ser ejecutado a la clase “*DBController”* específicamente a las funciones *“conectarDB”*, *“desconectarDB”* y *“ejecutarQuery”*, donde se buscará validar la correcta funcionalidad de estas.

Los errores evaluados para esta prueba son:

* ER01.-No conexión a base de datos.
* ER02.-No desconexión a base de datos.
* ER03.-No persistencia de conexión.

Cada caso de evaluación se encuentran descritos en [2], cada tester deberá ejecutar el ***unit test*** programado, ingresar entradas correspondientes al caso y presentar resultados en el documento.

***PU16.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

***Altas de registros***

*PU17.- Altas de Pacientes*

***PU17.01.- Caja Negra:*** ***basada en fallas***

Para la elaboración de esta prueba será necesario el desarrollo de una herramienta de trabajo, la cual cree de manera dinámica paquetes de datos para su utilización en la prueba.

La creación de estos datos considerará las especificaciones presentadas en el “Diccionario de Datos”[3] del proyecto, estas son:

* + Tipo de dato
  + Longitud
  + Formatos y/o restricciones
  + Dominio

Los paquetes de datos serán generados con el fin de probar las posibles combinaciones por cada campo, dichas combinaciones se realizarán en base a los casos de error y éxito de la validación de campos.

Los posibles casos de error considerados para la prueba por cada campo son:

* ER01.- No cumple “not null”
* ER02.- No cumple con tamaño
* ER03.- No cumple tipo de dato
* ER04.- No cumple formato
* ER05.- No cumple con valores aceptados (Dominios)
* ER06.- No cumple con obligación

En cada set de datos es importante considerar de manera general los siguientes errores:

* ER07.- No existe el campo

Una vez definidos los errores a considerar se analizó la ocurrencia de cada uno de ellos, presentados en la Tabla 2, siendo “1” como posible error en la inserción de información a el campo y “0” como no presente.

Tabla 2 Documento “Paciente” posibles errores.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Campo** | **ER01** | **ER02** | **ER03** | **ER04** | **ER05** | **ER07** |
| nombre | 1 | 1 | 1 | 1 | 0 | 0 |
| apellidoPat | 1 | 1 | 1 | 1 | 0 | 0 |
| apellidoMat | 0 | 1 | 1 | 1 | 0 | 0 |
| sexo | 1 | 1 | 0 | 0 | 1 | 0 |
| fechaNacimiento | 1 | 0 | 1 | 1 | 0 | 0 |
| lugarNacimiento | 0 | 1 | 1 | 1 | 0 | 0 |
| ocupacion | 0 | 0 | 0 | 0 | 1 | 0 |
| escolaridad | 0 | 0 | 0 | 0 | 1 | 0 |
| religiones | 0 | 0 | 0 | 0 | 1 | 0 |
| edoCivil | 0 | 0 | 0 | 0 | 1 | 0 |
| telefono | 1 | 1 | 1 | 1 | 0 | 0 |
| correo | 1 | 1 | 1 | 1 | 0 | 0 |
| fechaRegistro | 1 | 1 | 1 | 1 | 0 | 0 |
| \*Errores generales | 0 | 0 | 0 | 0 | 0 | 1 |

\*Los errores generales se consideran aplicados como un campo extra.

*Fuente: Elaboración propia.*

De acuerdo con todas las combinaciones posibles de interacción de errores serán creados los casos para los escenarios incorrectos, mientras que para los casos correctos se generarán datos aceptados por la base de datos. Estos paquetes serán divididos entre el número de casos considerados en el reporte de pruebas, con el fin de probar cada uno de los datos generados y evitar la presencia de un mismo error en los casos de escenario.

Al concluir la generación de los datos se procederá a cargar de manera automática los paquetes, mediante un ***unit test***, con el cual se comprobara el correcto funcionamiento de la función *“agregarPaciente()”*, registrando por cada caso a evaluar el resultado de la inserción (Exitosa o Fallida), se realizará una evaluación comparativa para conocer la efectividad de la prueba, comparando el estado del caso (Incorrecto o Correcto) con el resultado de la inserción (Exitosa o Fallida) considerando que los casos incorrectos deberán retornar inserciones fallidas y los correctos exitosas. Los resultados y observaciones deberán ser redactados en el reporte designado para la prueba en [4].

Es importante mencionar que para estas pruebas solo se ejecutara en las bases de datos existentes en el dispositivo, debido a la ejecución de pruebas anteriores.

***PU17.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU18.- Altas de Historial Clínico*

***PU18.01.- Caja Negra:*** ***basada en fallas***

Tal y como se mencionó en la definición de la prueba PU17.01 esta prueba está diseñada de la misma manera, considerando las mismas características de interés en los datos, así como lo errores. Debido a la complejidad en la que está construido el Documento HistorialClinico es que se optó por presentar las tablas de incidencias de error por cada documento embebido que lo conforma. Además, es importante mencionar que para este caso las combinaciones posibles de incidencia de errores fueron calculadas por niveles, y posteriormente sumadas para cada documento.

La Tabla 3 muestra el mapeo por campos, siendo “1” como posible error en la inserción de información a el campo y “0” como no presente.

Tabla 3 Documento "HistorialClinico" posibles errores.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Campo** | **ER01** | **ER02** | **ER03** | **ER04** | **ER05** | **ER07** |
| paciente\_id | 1 | 0 | 0 | 0 | 1 | 0 |
| tipo | 1 | 0 | 0 | 0 | 1 | 0 |
| fechaCambio | 1 | 0 | 0 | 1 | 0 | 0 |
| Antecedentes | 0 | 0 | 1 | 0 | 0 | 0 |
| Heredofamiliar | 0 | 0 | 1 | 0 | 0 | 0 |
| NoPatologico | 0 | 0 | 1 | 0 | 0 | 0 |
| Patologico | 0 | 0 | 1 | 0 | 0 | 0 |
| \*Errores generales | 0 | 0 | 0 | 0 | 0 | 1 |

\*Los errores generales se consideran aplicados como un campo extra.

*Fuente: Elaboración propia.*

La Tabla 4 muestra el mapeo por campos del documento embebido “Heredofamiliares”, siendo “1” como posible error en la inserción de información a el campo y “0” como no presente.

Tabla 4 Documento Embebido "Heredofamiliares" posibles errores.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Campo** | **ER01** | **ER02** | **ER03** | **ER04** | **ER05** | **ER07** |
| padecimientos | 1 | 0 | 1 | 0 | 0 | 0 |
| nombre | 1 | 0 | 0 | 0 | 0 | 0 |
| estado | 1 | 0 | 1 | 0 | 0 | 0 |
| \*Errores generales | 0 | 0 | 0 | 0 | 0 | 1 |

\*Los errores generales se consideran aplicados como un campo extra.

*Fuente: Elaboración propia.*

La Tabla 5 muestra el mapeo por campos del documento embebido “NoPatologicos”, siendo “1” como posible error en la inserción de información a el campo y “0” como no presente.

Tabla 5 Documento Embebido "NoPatologicos" posibles errores.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Campo** | **ER**  **01** | **ER**  **02** | **ER**  **03** | **ER**  **04** | **ER**  **05** | **ER**  **07** | | aseoDiario | 0 | 0 | 0 | 0 | 1 | 0 | | higieneDental | 0 | 0 | 0 | 0 | 1 | 0 | | Dietético | 1 | 0 | 1 | 0 | 0 | 0 | | apetito | 0 | 0 | 0 | 0 | 1 | 0 | | mayorApetito | 0 | 0 | 0 | 0 | 1 | 0 | | comidas | 1 | 1 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 0 | 0 | | estado | 1 | 0 | 1 | 0 | 0 | 0 | | cantidad | 1 | 1 | 0 | 1 | 0 | 0 | | dondeCome | 1 | 0 | 0 | 0 | 1 | 0 | | compania | 1 | 0 | 0 | 0 | 1 | 0 | | preparador | 1 | 0 | 0 | 0 | 1 | 0 | | comerMismaHora | 1 | 0 | 1 | 0 | 0 | 0 | | saltarseComidas | 1 | 0 | 1 | 0 | 0 | 0 | | tipoAlimentos | 0 | 1 | 0 | 1 | 0 | 0 | | preparaciones | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | estado | 1 | 0 | 1 | 0 | 0 | 0 | | grasas | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | estado | 1 | 0 | 1 | 0 | 0 | 0 | | sal | 1 | 0 | 1 | 0 | 0 | 0 | | alimentosIntolerantes | 0 | 1 | 0 | 1 | 0 | 0 | | alimentosNoAgrado | 0 | 1 | 0 | 1 | 0 | 0 | | alimentosApetecen | 0 | 1 | 0 | 1 | 0 | 0 | | animo | 1 | 0 | 1 | 0 | 0 | 0 | | modificacion | 1 | 0 | 1 | 0 | 0 | 0 | | causa | 0 | 1 | 0 | 1 | 0 | 0 | | liquidos | 1 | 0 | 1 | 0 | 0 | 0 | | tipo | 1 | 0 | 0 | 0 | 1 | 0 | | cantidad | 1 | 0 | 0 | 0 | 1 | 0 | | agregado | 1 | 0 | 1 | 0 | 0 | 0 | | alcohol | 1 | 0 | 1 | 0 | 0 | 0 | | tabaco | 1 | 0 | 1 | 0 | 0 | 0 | | medicamentos | 0 | 1 | 0 | 1 | 0 | 0 | | suplemento | 1 | 0 | 1 | 0 | 0 | 0 | | suplementos | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 1 | 0 | 1 | 0 | 0 | | dosis | 1 | 1 | 0 | 1 | 0 | 0 | | frecuenciasConsumo | 1 | 0 | 1 | 0 | 0 | 0 | | grupo | 1 | 0 | 0 | 0 | 1 | 0 | | diario | 0 | 1 | 0 | 1 | 0 | 0 | | semanal | 1 | 1 | 0 | 1 | 0 | 0 | | quincenal | 0 | 1 | 0 | 1 | 0 | 0 | | alimentos | 0 | 1 | 0 | 1 | 0 | 0 | | valoracionNutricional | 0 | 1 | 0 | 1 | 0 | 0 | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Campo** | **ER**  **01** | **ER**  **02** | **ER**  **03** | **ER**  **04** | **ER**  **05** | **ER**  **07** | | Deportivos | 0 | 0 | 1 | 0 | 0 | 0 | | edadAF | 1 | 1 | 0 | 1 | 0 | 0 | | deportesAnteriores | 0 | 0 | 0 | 1 | 0 | 0 | | sedentario | 1 | 0 | 1 | 0 | 0 | 0 | | imposibilidades | 0 | 1 | 0 | 1 | 0 | 0 | | malestares | 0 | 1 | 0 | 1 | 0 | 0 | | liquidos | 1 | 0 | 1 | 0 | 0 | 0 | | tipo | 1 | 1 | 0 | 1 | 0 | 0 | | cantidad | 1 | 1 | 0 | 1 | 0 | 0 | | desidratacion | 1 | 0 | 1 | 0 | 0 | 0 | | habitual | 1 | 0 | 1 | 0 | 0 | 0 | | deportesActuales | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 1 | 0 | 1 | 0 | 0 | | faseEntto | 1 | 0 | 0 | 0 | 1 | 0 | | horaDia | 1 | 1 | 0 | 1 | 0 | 0 | | modalidad | 0 | 1 | 0 | 1 | 0 | 0 | | posiciones | 0 | 1 | 0 | 1 | 0 | 0 | | \*Errores generales | 0 | 0 | 0 | 0 | 0 | 1 | |

\*Los errores generales se consideran aplicados como un campo extra.

*Fuente: Elaboración propia.*

La Tabla 6 muestra el mapeo por campos del documento embebido “Patologicos”, siendo “1” como posible error en la inserción de información a el campo y “0” como no presente.

Tabla 6 Documento Embebido 'Patológicos" posibles errores.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Campo** | **ER01** | **ER02** | **ER03** | **ER04** | **ER05** | **ER07** |
| grupoSanguineo | 1 | 0 | 0 | 0 | 1 | 0 |
| rh | 1 | 0 | 0 | 0 | 1 | 0 |
| transfusion | 1 | 0 | 1 | 0 | 0 | 0 |
| motivo | 1 | 1 | 0 | 1 | 0 | 0 |
| periodoTransfusion | 1 | 1 | 0 | 1 | 0 | 0 |
| desmayo | 1 | 0 | 1 | 0 | 0 | 0 |
| operacion | 1 | 0 | 1 | 0 | 0 | 0 |
| operaciones | 0 | 1 | 0 | 1 | 0 | 0 |
| enfermedades | 1 | 0 | 1 | 0 | 0 | 0 |
| nombre | 1 | 0 | 0 | 0 | 1 | 0 |
| estado | 1 | 0 | 1 | 0 | 1 | 0 |
| internamiento | 1 | 0 | 1 | 0 | 1 | 0 |
| lesiones | 0 | 0 | 1 | 0 | 0 | 0 |
| area | 1 | 0 | 0 | 0 | 1 | 0 |
| estado | 1 | 0 | 1 | 0 | 0 | 0 |
| alergia | 1 | 0 | 1 | 0 | 0 | 0 |
| alergMedicamentos | 0 | 1 | 0 | 1 | 0 | 0 |
| alergAlimentos | 0 | 1 | 0 | 1 | 0 | 0 |
| alergOtros | 0 | 1 | 0 | 1 | 0 | 0 |
| Ginecobstetrico | 0 | 0 | 1 | 0 | 1 | 0 |
| inicioMenarca | 1 | 0 | 1 | 0 | 0 | 0 |
| edadInicio | 1 | 0 | 0 | 1 | 0 | 0 |
| gestacion | 1 | 0 | 0 | 1 | 0 | 0 |
| aborto | 1 | 0 | 0 | 1 | 0 | 0 |
| cesarea | 1 | 0 | 0 | 1 | 0 | 0 |
| regularidad | 1 | 0 | 1 | 0 | 0 | 0 |
| \*Errores generales | 0 | 0 | 0 | 0 | 0 | 1 |

\*Los errores generales se consideran aplicados como un campo extra.

*Fuente: Elaboración propia.*

La prueba será realizada una vez se hayan creado los paquetes de datos necesarios para la misma, entregando un juego de ellos a cada tester para iniciar y realizar el reporte correspondiente.

***PU18.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU19.- Altas de Consulta*

***PU19.01.- Caja Negra:*** ***basada en fallas***

La siguiente prueba ha sido definida de la misma manera que las pruebas PU17.01 y PU18.01, considerando nuevamente las características de los datos y los errores mencionados anteriormente. Al igual que en la prueba anterior las combinaciones posibles de incidencia de errores fueron calculadas por niveles, y posteriormente sumadas para el documento.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Campo** | **ER**  **01** | **ER**  **02** | **ER**  **03** | **ER**  **04** | **ER**  **05** | **ER**  **07** | | fecha | 1 | 0 | 1 | 1 | 0 | 0 | | historial\_id | 1 | 0 | 0 | 0 | 1 | 0 | | sintomas | 0 | 1 | 0 | 1 | 0 | 0 | | tratamiento | 1 | 0 | 1 | 0 | 0 | 0 | | tratamientos | 1 | 1 | 0 | 1 | 0 | 0 | | signosVitales | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | magnitud | 1 | 0 | 1 | 0 | 0 | 0 | | exporacionesFisicas | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | observación | 1 | 1 | 0 | 1 | 0 | 0 | | Antropometria | 1 | 0 | 1 | 0 | 0 | 0 | | medidas | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | magnitud | 1 | 1 | 1 | 0 | 0 | 0 | | nivel | 1 | 0 | 0 | 0 | 1 | 0 | | diagnosticos | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | magnitud | 1 | 1 | 1 | 0 | 0 | 0 | | formula | 1 | 1 | 0 | 1 | 0 | 0 | | indicador | 1 | 1 | 0 | 1 | 0 | 0 | | composicionesCorporales | 1 | 0 | 1 | 0 | 0 | 0 | | tipo | 1 | 0 | 0 | 0 | 1 | 0 | | porcentaje | 1 | 1 | 1 | 0 | 0 | 0 | | peso | 1 | 1 | 1 | 0 | 0 | 0 | | formula | 1 | 1 | 0 | 1 | 0 | 0 | | drinkwater | 1 | 1 | 1 | 0 | 0 | 0 | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Campo** | **ER**  **01** | **ER**  **02** | **ER**  **03** | **ER**  **04** | **ER**  **05** | **ER**  **07** | | PlanNutricional | 0 | 0 | 1 | 0 | 0 | 0 | | tipoDieta | 1 | 0 | 0 | 1 | 0 | 0 | | observaciones | 0 | 1 | 0 | 1 | 0 | 0 | | calculodietetico | 1 | 0 | 1 | 0 | 0 | 0 | | nombre | 1 | 0 | 0 | 0 | 1 | 0 | | magnitud | 1 | 1 | 1 | 0 | 0 | 0 | | recordatorios | 0 | 0 | 1 | 0 | 0 | 0 | | tiempo | 1 | 0 | 0 | 0 | 1 | 0 | | horario | 0 | 0 | 1 | 1 | 0 | 0 | | alimentos | 1 | 0 | 1 | 0 | 0 | 0 | | alimento | 1 | 1 | 0 | 1 | 0 | 0 | | cantidad | 1 | 1 | 0 | 1 | 0 | 0 | | porción | 1 | 1 | 0 | 1 | 0 | 0 | | pt | 1 | 1 | 0 | 1 | 0 | 0 | | lip | 1 | 1 | 0 | 1 | 0 | 0 | | hco | 1 | 1 | 0 | 1 | 0 | 0 | | dietosinteticos | 1 | 0 | 1 | 0 | 0 | 0 | | grupo | 1 | 0 | 0 | 0 | 1 | 0 | | gramos | 1 | 1 | 1 | 0 | 0 | 0 | | calorías | 1 | 1 | 1 | 0 | 0 | 0 | | porcentajes | 1 | 1 | 1 | 0 | 0 | 0 | | planes | 0 | 0 | 1 | 0 | 0 | 0 | | tiempo | 1 | 0 | 0 | 0 | 1 | 0 | | horario | 0 | 0 | 1 | 1 | 0 | 0 | | opciones | 1 | 1 | 0 | 1 | 0 | 0 | | \*Errores generales | 0 | 0 | 0 | 0 | 0 | 1 | |

\*Los errores generales se consideran aplicados como un campo extra.

*Fuente: Elaboración propia.*

***PU19.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

***Bajas de registros***

*PU20.-Bajas de Pacientes*

***PU20.01.- Caja Negra:*** ***basada en fallas***

En esta prueba se realizará mediante la utilización de un **unit test** con el fin de validar la correcta funcionalidad de la función “eliminarPaciente” dentro de la clase “DBController”, para ello se han planteado casos de prueba de escenarios, con entradas establecidas para este test y resultados esperados. Es importante mencionar que debido a la lógica de operaciones establecida para SICMA los registros no son eliminados, son considerados como borrados lógicos, presentado dos estados “activo” e “inactivo”, en base a ellos la aplicación prepara las funciones posteriores de utilización.

***PU20.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

***Cambios en registros***

*PU21.- Cambios de Pacientes*

***PU21.01.- Caja Negra:*** ***basada en fallas***

Para la elaboración de esta prueba será necesario la utilización de los paquetes de datos generados en pruebas anteriores. Mediante la ejecución de un ***unit test*** los datos serán cargados realizando solicitudes de cambios a cada campo, considerando las características de datos, así como lo posibles errores de las pruebas de altas de registro.

Dentro de este test no se ejecutará el paquete de datos de manera con el que fue creado, con el fin de validar los cambios mediante la comparación de datos ingresados y datos presentes en la base y de esta forma validar igualmente la efectividad del proceso.

Cada caso de evaluación se encuentran descritos en [2], cada tester deberá ejecutar el ***unit test*** programado, ingresar entradas correspondientes al caso y presentar resultados en el documento.

***PU21.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU22.- Cambios de Historial Clínico*

***PU22.01.- Caja Negra:*** ***basada en fallas***

Para la elaboración de esta prueba será necesario la utilización de los paquetes de datos generados en pruebas anteriores. Mediante la ejecución de un ***unit test*** los datos serán cargados realizando solicitudes de cambios a cada campo, considerando las características de datos, así como lo posibles errores de las pruebas de altas de registro.

Dentro de este test no se ejecutará el paquete de datos de manera con el que fue creado, con el fin de validar los cambios mediante la comparación de datos ingresados y datos presentes en la base y de esta forma validar igualmente la efectividad del proceso.

Cada caso de evaluación se encuentran descritos en [2], cada tester deberá ejecutar el ***unit test*** programado, ingresar entradas correspondientes al caso y presentar resultados en el documento.

***PU22.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU23.- Cambios de Consultas*

***PU23.01.- Caja Negras:*** ***basada en fallas***

Para la elaboración de esta prueba será necesario la utilización de los paquetes de datos generados en pruebas de inserción. Mediante la ejecución de un ***unit test*** los datos serán cargados realizando solicitudes de cambios a cada campo, considerando las características de datos, así como lo posibles errores de las pruebas de altas de registro.

Dentro de este test no se ejecutará el paquete de datos de manera con el que fue creado, con el fin de validar los cambios mediante la comparación de datos ingresados y datos presentes en la base y de esta forma validar igualmente la efectividad del proceso.

Cada caso de evaluación se encuentran descritos en [2], cada tester deberá ejecutar el ***unit test*** programado, ingresar entradas correspondientes al caso y presentar resultados en el documento.

***PU23.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

**Consultas en registros**

*PU24.- Consulta de Pacientes*

***PU24.01.- Caja Negra:*** ***basada en fallas***

Para la realización de esta prueba será implementado un test mediante la creación de ***unit test***, aplicado a la función de consulta del documento, cuyo objetivo será comprobar la correcta funcionalidad de ésta. Las entradas y resultados esperados son presentados en el reporte correspondiente a la prueba. En la realización de estas pruebas se considera la ejecución de pruebas anteriores, tal como la de PU20.-Baja de registros.

Los posibles casos de error considerados para la prueba son:

* ER01.-No existencia de id\_ de elemento a buscar.
* ER02.-No existencia de elemento a buscar.

***PU24.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU25.- Consulta de Historial Clínico*

***PU25.01.- Caja Negra:*** ***basada en fallas***

Para la realización de esta prueba será implementado un test mediante la creación de ***unit test***, aplicado a la función de consulta del documento, cuyo objetivo será comprobar la correcta funcionalidad de ésta. Las entradas y resultados esperados son presentados en el reporte correspondiente a la prueba. En la realización de estas pruebas se considera la ejecución de pruebas anteriores, tal como la de PU20.-Baja de registros. Con respecto a la lógica de operaciones planteada en SICMA, esta prueba no proporcionara información detallada sobre las “Consultas”, únicamente su fecha de realización.

Los posibles casos de error considerados para la prueba son:

* ER01.-No existencia de id\_ de elemento a buscar.
* ER02.-No existencia de elemento a buscar.

***PU25.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU26.- Consulta de Consultas*

***PU26.01.- Caja Negra:*** ***basada en fallas***

Para la realización de esta prueba será implementado un test mediante la creación de ***unit test***, aplicado a la función de consulta del documento, cuyo objetivo será comprobar la correcta funcionalidad de ésta. Las entradas y resultados esperados son presentados en el reporte correspondiente a la prueba. En la realización de estas pruebas se considera la ejecución de pruebas anteriores, tal como la de PU20.-Baja de registros. Estas pruebas consideran únicamente el retorno de información correspondiente a historial clínico.

Los posibles casos de error considerados para la prueba son:

* ER01.-No existencia de id\_ de elemento a buscar.
* ER02.-No existencia de elemento a buscar.

***PU26.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

*PU27.- Eliminación de base de datos local en dispositivo*

***PU27.01.- Caja Negra:*** ***basada en fallas***

La prueba desarrollada para la comprobación de la creación correcta de la base de datos se hará uso de las herramientas proporcionadas por Flutter para la realización de pruebas, se desarrollará un ***unit test*** para ser ejecutado a la clase “*DBController”* específicamente a la función *“eliminarDB”*, donde se buscará validar la correcta funcionalidad de esta.

Los posibles casos de error considerados para la prueba son:

* ER01.-No existencia de nombre de base de datos.
* ER03.-No existencia de usuario de base de datos.
* ER04.-No existencia de contraseña de base de datos.
* ER05.-No eliminación de base de datos.
* ER06.- Datos de Autorización incorrectos
* ER07.-No existencia de base de datos

Cada caso de evaluación se encuentran descritos en [2], cada tester deberá ejecutar el ***unit test*** programado y presentar resultado en el documento. Cabe mencionar que esta prueba depende de la ejecución previa de la prueba *PU15.- Creación de base de datos.*

***PU27.02.- Caja Blanca: Análisis estático de código***

Dart, el lenguaje de programación a utilizar, tiene una herramienta de análisis estático, éste le permite encontrar problemas antes de ejecutar una sola línea de código. La cual será utilizada para encontrar posibles errores y de esta manera garantizar que el código cumpla con las pautas de formato, tipo de dato, utilización, entre otras.

**MD02.01.-** **Sistema embebido**

**CM06.01.- Sensor angular**

*PU28.-Funcionalidad*

Prueba establecida para que el sensor angular marque la resistencia correcta mediante la medición con un multímetro.

*PU29.- Precisión*

Prueba establecida para la validación de la medición de la resistencia tomando en cuenta la edición de un plicómetro.

**CM07.01.- Arduino**

*PU30.- Funcionalidad*

Prueba para la verificación de la funcionalidad del dispositivo Arduino ,tomando en cuenta la funcionalidad y compilación de código sobre el mismo.

*PU31.- Comunicación con módulo bluetooth*

*PU32.- Envío de datos con módulo bluetooth*

*PU33.- Comunicación con sensor angular*

**CM08.01.- Módulo bluetooth**

*PU34.- Funcionalidad*

Prueba para la verificación de la funcionalidad del módulo bluetooth, tomando en cuenta la funcionalidad y uso del módulo mediante Arduino.

*PU35.- Envío de datos.*

Prueba para la validación de la integridad de los datos, enviados por el módulo bluetooth de Arduino.

**MD03.01.- Google Drive**

**CM09.01.- Respaldo y restauración**

*PU36.- Comunicación con el servicio*

Prueba para validar la comunicación y la correcta conexión con el servicio.

*PU37.- Respaldo de información*

Prueba para validar la integridad de datos respaldados en el servicio de Google.

*PU38.- Recuperación de información*

Prueba para validar la integridad de datos recuperados desde el servicio de Google.

Pruebas Integración

**Índice de contenido**

[**MD01.-** **Aplicación móvil** 39](#_Toc55439506)

[PI-CM01 Sistema de visión artificial 39](#_Toc55439507)

[PI-CM02 Sistema de reconocimiento del cuerpo humano 39](#_Toc55439508)

[PI-CM03 Interfaz gráfica de usuario 39](#_Toc55439509)

[PI-CM04 Modulo bluetooth 39](#_Toc55439510)

[PI-CM05 Base de datos 39](#_Toc55439511)

**[MD02.- Sistema Embebido](#_Toc55439512)** [39](#_Toc55439512)

[PI-CM06 Sensor angular 39](#_Toc55439513)

[PI-CM07 Arduino 39](#_Toc55439514)

[PI CM08 Módulo bluetooth 39](#_Toc55439515)

[**MD03.- Google Drive** 39](#_Toc55439516)

[PI-CM09 Respaldo y restauración 40](#_Toc55439517)

**MD01.-** **Aplicación móvil**

PI-CM01 Sistema de visión artificial

PI-CM02 Sistema de reconocimiento del cuerpo humano

PI-CM03 Interfaz gráfica de usuario

PI-CM04 Modulo bluetooth

PI-CM05 Base de datos

**MD02.- Sistema Embebido**

PI-CM06 Sensor angular

PI-CM07 Arduino

PI PI-CM08 Módulo bluetooth

**MD03.- Google Drive**

PI-CM09 Respaldo y restauración

Pruebas de Sistema

PS-MD02.-Prueba de Hardware

PS-MD03.-Prueba de Software