Profesores:

Pedro Renedo Fernández

**Grado en Ingeniería Informática**

Asignatura: Validación y Pruebas (Online)

Materia: Lenguajes y Sistemas Informáticos

Departamento: Ingeniería Informática



García Armario, Jesus

Documentación:

**TDD-ACB-JGA**



Cordero Bernal, Adrián

Pedro Renedo Fernández

Revisado por:

**2022/23**

**Contenido**

[1. Metodología 3](#_Toc122178140)

[2. Requisitos Funcionales 3](#_Toc122178141)

[3. Metodología de Diseño 4](#_Toc122178142)

[3.1. Product Backlog 5](#_Toc122178143)

[3.2. Sprint Backlog 7](#_Toc122178144)

[4. Historia Gráfica 8](#_Toc122178145)

[4.1. Sprint 1 8](#_Toc122178146)

[4.2. Sprint 2 18](#_Toc122178147)

[5. Diagrama de paquetes del sistema 25](#_Toc122178148)

[6. Diagrama de clases de diseño. 25](#_Toc122178149)

[7. Proyecto de Prueba 25](#_Toc122178150)

[7.1. Pruebas Unitarias 25](#_Toc122178151)

[7.2. Pruebas de Integración 25](#_Toc122178152)

[8. Conclusiones 25](#_Toc122178153)

# Metodología

Para la elaboración de este proyecto de prácticas, se ha llevado a cabo una metodología **ágil**, inspirada en SCRUM, y basada en incrementos de la aplicación basada en los requisitos funcionales.

El desarrollo ha estado implementado mediante una estrategia *DDT* (tests guiados por datos) y *TDD* (desarrollo guiado por pruebas).

De manera colaborativa, hemos hecho uso de un repositorio compartido, especificando tareas en *Product* Backlog, hasta poder confeccionar un *Sprint Backlog*, publicando las modificaciones del proyecto a través de una plataforma *git*.

El proyecto está alojado en *GitHub*, y es accesible en: <https://github.com/Adripy/TDD-ACB-JGA>

# Requisitos Funcionales

|  |  |  |
| --- | --- | --- |
| Código | Descripción | Función |
| RF1 | Se desea crear un sistema de validación de datos personales. | Valida. |
| RF2 | Se desea crear un sistema de cálculo de estadísticos. | Estadística. |
| RF3 | Se desea que el sistema de validación compruebe si un código postal es válido. | Validación de CP |
| RF4 | Se desea que la validación del CP devuelva la provincia a la que pertenece. | Validación de CP |
| RF5 | Se desea que el sistema compruebe si un NIF es válido. | Validación de NIF |
| RF6 | Se desea que la validación de NIF devuelva el tipo. | Validación de NIF. |
| RF7 | Se desea validar una tarjeta de crédito de tipo VISA o Mastercard. | Validación de Tarjeta. |
| RF8 | Se desea validar un CCC bancario. | Validación de CCC. |
| RF9 | Se desea validar un IBAN español. | Validación de IBAN. |
| RF10 | Se desea que el sistema sea capaz de calcular la media aritmética de una lista de valores. | Cálculo de Media Aritmética. |
| RF11 | Se desea que el sistema calcule la media geométrica de un conjunto de valores*.* | Cálculo de Media Geométrica. |
| RF12 | Se desea que el sistema calcule la media armónica de un conjunto de valores numéricos. | Cálculo de Media Armónica. |
| RF13 | Se desea que el sistema calcule la mediana entre una lista de valores. | Cálculo de Mediana. |
| RF14 | Se desea que el sistema calcule la moda entre una lista de valores. | Cálculo de Moda. |
| RF15 | Se desea que el sistema calcule la desviación absoluta de los valores numéricos. | Cálculo de Desviación Absoluta. |
| RF16 | Se desea que el sistema calcule la desviación media de un conjunto de valores numéricos. | Cálculo de Desviación Media. |

# Metodología de Diseño

Se ha optado por una organización ágil, basada en SCRUM. Contamos con un repertorio de tareas, que conforman el Product BackLog, que hemos dividido en dos Sprints, el primero centrado en el desarrollo de la clase Estadística y el segundo en el de la clase Valida.

## Product Backlog

Imagen que contiene Gráfico

Descripción generada automáticamenteAccesible en nuestro repositorio Github en: <https://github.com/Adripy/TDD-ACB-JGA/issues>

Tabla

Descripción generada automáticamente con confianza media

## Sprint Backlog

A medida que las tareas conformaban suficientes para desarrollar un entregable, éstas se añadían al Sprint Backlog.

Se configuraron dos Sprints en esta tarea:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

# Historia Gráfica

## Sprint 1

ROJO

|  |  |
| --- | --- |
| Imágenes | Fecha |
| Captura de pantalla de computadora  Descripción generada automáticamente | 14/12/2022 -> 17:15 |
|  | 14/12/2022 -> 17:47 |
|  | 14/12/2022 -> 17:54 |
|  | 14/12/2022 -> 17:57 |
|  | 14/12/2022 -> 18:10 |
|  | 14/12/2022 -> 18:24 |
|  | 14/12/2022 -> 18:35 |
|  | 14/12/2022 -> 18:43 |
|  | 14/12/2022 -> 20:01 |

VERDE

|  |  |
| --- | --- |
| Imágenes | Fecha |
|  | 14/12/2022 -> 20:01 |
|  | 14/12/2022 -> 20:12 |
|  | 14/12/2022 -> 20:15 |
|  | 14/12/2022 -> 20:27 |
|  | 14/12/2022 -> 20:56 |
|  | 14/12/2022 -> 21:00 |
|  | 15/12/2022 -> 09:47 |

REFACTORIZAR

|  |  |
| --- | --- |
| Imágenes | Fecha |
|  | 15/12/2022 -> 09:49 |
|  | 15/12/2022 -> 10:15 |
|  | 15/12/2022 -> 10:17 |

## Sprint 2

ROJO

|  |  |
| --- | --- |
| Imágenes | Fecha |
|  | 15/12/2022 -> 11:12 |
|  | 15/12/2022 -> 11:35 |
|  | 15/12/2022 -> 12:00 |
|  | 15/12/2022 -> 12:16 |
|  | 15/12/2022 -> 12:30 |
|  | 15/12/2022 -> 17:01 |

VERDE

|  |  |
| --- | --- |
| Imágenes | Fechas |
|  | 15/12/2022 -> 18:29 |
|  | 15/12/2022 -> 18:53 |
|  | 15/12/2022 -> 19:55 |
|  | 15/12/2022 -> 22:35 |
|  | 15/12/2022 -> 23:01 |
|  | 15/12/2022 -> 00:09 |
|  | 16/12/2022 -> 10:00 |
|  | 16/12/2022 -> 10:00 |

REFACTORIZAR

# Diagrama de paquetes del sistema

# Diagrama de clases de diseño.

# Proyecto de Prueba

## Pruebas Unitarias

Se encuentran codificadas y comentadas, con cada caso de prueba, en las clases *ClasesLibTests* y *ComunTests.*

## Pruebas de Integración

Se encuentran codificadas y comentadas, con los casos de prueba, en la clase *BaseDatosTests*.

# Conclusiones