# Tests:

## Paquet data

### Classe NIF

En la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe NIF. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 6 objectes diferents. Un amb un NIF correcte, i la resta incorrectes. Amb més números del compte, sense cap caràcter al final, amb caràcters de sobra, amb l’ordre alterat de lletres i números, i per últim un NIF buit.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

També hem realitzat un @Test per a la comprovació de les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el NIF es correcte (retorna Vertader o Fals depenent del cas).

### Classe AccredNumbTest

n la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe AccredNumb. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 5 objectes diferents. Un amb un Número d’Acreditació correcte, i la resta incorrectes. No havent número d’acreditació, amb més i amb menys números del compte, i sent els caràcters en comptes de números.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

També hem realitzat un @Test per a la comprovació de les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el Número d’Acreditació es correcte (retorna Vertader o Fals depenent del cas).

### Classe DocPathTest

En la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe DocPath. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 2 objectes diferents. Un amb una ruta d’un document, i una altre inicialitzat a null.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

També hem realitzat un @Test per a la comprovació de les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el Número d’Acreditació es correcte (retorna Vertader o Fals depenent del cas).

### Classe EncryptedDataTest

En la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe DocPath. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 3 objectes diferents. Dos d’aquests són correctes i l’altre es null.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

Un altre @Test comprova les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el Número d’Acreditació es correcte (retorna Vertader o Fals depenent del cas).

I per últim tenim un @Test que el que fa es comprovar que la funció toString que hem implementat funcioni correctament.

### Classe EncryptingKeyTest

En la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe EncryptingKey. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 2 objectes diferents. Un de correcte, sent aquest una clau d’encriptació, i l’altre es incorrecte, sent aquest null.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

Un altre @Test comprova les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el Número d’Acreditació es correcte (retorna Vertader o Fals depenent del cas).

### Classe PINcodeTest

En la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe PINcode. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 5 objectes diferents. Un amb un codi PIN correcte, i la resta incorrectes. Sent un d’ells null, també amb més i amb menys dígits del compte, i sent els caràcters en comptes de números.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

També hem realitzat un @Test per a la comprovació de les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el Número d’Acreditació es correcte (retorna Vertader o Fals depenent del cas).

### 

### Classe PasswordTest

En la realització d’aquest test en concret es comprova si es realitzen correctament les excepcions de la implementació de la classe Password. Per a poder realitzar aquestes comprovacions es necessari un @BeforeAll, on s’inicialitzaran 5 objectes diferents. Un amb un password correcte, i la resta incorrectes. Sent un d’ells null, amb menys dígits del compte, essent format solament de números i estant format solament de caràcters.

Una vegada inicialitzats els objectes anteriors, realitzem un @Test per a comprovar que la funció equals que hem implementat funcioni correctament.

També hem realitzat un @Test per a la comprovació de les excepcions. Per a la realització d’aquesta tasca el que fem es cridar a la funció de comparació assertTrue/False que ens comprova si el Número d’Acreditació es correcte (retorna Vertader o Fals depenent del cas).

## Paquet Services

No cal passar tests a les classes substitutes (CertificationAuthorityImplTest i SSImplTest), ni tampoc fer aquestes classes massa elaborades. Encara així com a extra les hem realitzat amb una petita elaboració per facilitar la seva comprovació i entendre els seus funcionaments

### Classe CertificationAuthorityImplTest

Realitzem un @BeforeEach on inicialitzem els objectes. Hem decidit que era convenient utilitzar un mètode de SetUp (@BeforeEach) per tal de preparar les classes a ser utilitzades i la resta de passos necessaris per iniciar els tests, evitant així repetir línies de codi.

A continuació tenim 5 @Test on hem implementat la comprovació de les funcions sendPIN, checkPin, checkCredent, sendCertfAuth i sendCertfAuthErrores de la classe CertificationAuthorityImpl, amb els seus respectius assertEquals i assertThrows, per a comprovar les excepcions.

### Classe SSImplTest

Realitzem un @BeforeEach on inicialitzem els objectes. Hem decidit que era convenient utilitzar un mètode de SetUp (@BeforeEach) per tal de preparar les classes a ser utilitzades i la resta de passos necessaris per iniciar els tests, evitant així repetir línies de codi.

A continuació tenim 2 @Test on hem implementat la comprovació de les funcions getLaboralLife i getMembAccreed de la classe SSImpl, amb els seus respectius assertEquals i assertThrows, per a comprovar les excepcions.

## Paquet publicadministration

### Classe QuotePeriodCollTest

Realitzem un @BeforeEach on inicialitzem els objectes. A continuació tenim 3 @Test on hem implementat la comprovació de les funcions addPeriod, addPeriodThrows i getInit de la classe QuotePeriodColl. Al @Test de addPeriod fem les comprovacions necessàries amb assertEquals, al @Test addPeriodThrows comprovem les excepcions mitjançant assertThrows, i per últim tenim el @Test del getter getInit, que comprova que funcioni correctament.

### Classe QuotePeriodTest

Realitzem un @BeforeEach on inicialitzem els objectes. A continuació tenim 4 @Test on hem implementat la comprovació de les funcions equals, notEquals, getNumDays i getInitDay de la classe QuotePeriod. Als @Test equals i notEquals fem les comprovacions necessàries amb assertEquals, i als @Test getNumDays i getInitDay es realitzen els getters que el que fan es comprovar que les funcions implementades funcionin correctament i no es realitzin test falsos, mitjançant les funcions assertEquals/assertNotEquals, respectivament.

### Classe PDFDocumentTest

Realitzem un @BeforeEach on inicialitzem l’objecte. A continuació tenim 5 @Test on hem implementat la comprovació de les funcions open, move, getCreateDate, getPath i getFile de la classe PDFDocument. Als @Test open i move fem les comprovacions de les funcions implementades i en el cas de que no es pogués fer qualsevol de les 2 funcions, llançarà una Exception. Als @Test getCreateDate, getPath i getFile es realitzen els getters que el que fan es comprovar que les funcions implementades funcionin correctament i no es realitzin test falsos, mitjançant les funcions assertEquals/assertNotEquals, respectivament.

### Classe LaboralLifeDocTest / Classe MemberAccreditationDocTest

A les següents classes s'utilitza una estructura semblant.

A cada classe s’inicialitzen al @BeforeEach els objectes pertinents i als @Test s’executen els seus respectius equals i getters, que s'estan utilitzant des de les diferents classes per comprovar que aquestes funcionen correctament.

La principal diferència entre les classes és que en MemberAccreditationDocTest validem el @Test mitjançant el nombre d’acreditació, i en canvi a LaboralLifeDocTest es valida mitjançant quote periods.