

**Facultad de Ingeniería**

Ingeniería de Software

**PROYECTO INGENIERIA SOFTWARE SCRUM++ 2**

**PARA SOFTWARE PARA CONTAPLUS**

Autores:

Joaquín Bello Bailoni  
Álvaro Brizuela Benavides  
Jairo Carrasco Garati  
Daniel Díaz Aravena  
Gabriel Psijas Rivillo  
Edson Yañez Retamal  
  
Profesor:  
Paulo Luis Francisco Quinzacara Jofre  
  
Santiago, Chile  
2025

# **ÍNDICE DE CONTENIDOS**

[**ÍNDICE DE CONTENIDOS 2**](#_heading=h.k516z6vyokc4)

[**ÍNDICE DE FIGURAS 4**](#_heading=h.2ysda09m44cx)

[**ÍNDICE DE TABLAS 7**](#_heading=h.henznhmwzn5e)

[**1. INTRODUCCIÓN 10**](#_heading=h.qql8w7f98v8b)

[**2. PROPUESTA DE SOLUCIÓN E INCREMENTO 11**](#_heading=h.mmw9v3nn1km5)

[**3. CASOS DE USOS SELECCIONADOS 13**](#_heading=h.5o150tihc5ak)

[**4. TABLA BACKLOG SIN TIEMPO REAL 73**](#_heading=h.f5l715nhceva)

[**5. DIAGRAMA DE CLASES 74**](#_heading=h.pt6ubw1uclfr)

[**6. DIAGRAMA DE DATOS 75**](#_heading=h.kpmznybbb2vq)

[**7. DIAGRAMA DE COMPONENTES 76**](#_heading=h.3s0ik8ssjmy4)

[**8. DIAGRAMA DE DESPLIEGUE 77**](#_heading=h.v0hhcq3u4bg8)

[**9. DIAGRAMA DE SECUENCIA 78**](#_heading=h.a76w9c3s6lgn)

[9.1 Caso de uso N°11 78](#_heading=h.kv8w5h2um89r)

[9.2 Caso de uso N°12 79](#_heading=h.qgfxkonr3mxd)

[9.3 Caso de uso N°13 80](#_heading=h.4fv8x4lfyefl)

[9.4 Caso de uso N°14 81](#_heading=h.ft06c4n6hb5c)

[9.5 Caso de uso N°15 82](#_heading=h.d98yu5kr1zqg)

[9.6 Caso de uso N°16 83](#_heading=h.of0sdir2d2u2)

[9.7 Caso de uso N°17 84](#_heading=h.3kjnzx89h30u)

[9.8 Caso de uso N°18 85](#_heading=h.q6vd28ym2zzx)

[9.9 Caso de uso N°19 86](#_heading=h.ij96edrtr81)

[9.10 Caso de uso N°20 87](#_heading=h.2cpalrkx9fjp)

[9.11 Caso de uso N°21 88](#_heading=h.u23g0vdk4afd)

[9.12 Caso de uso N°22 89](#_heading=h.jnf7ztmigk24)

[9.13 Caso de uso N°23 90](#_heading=h.2srubtxa4bb6)

[9.14 Caso de uso N°24 91](#_heading=h.dundakiubkma)

[9.15 Caso de uso N°25 92](#_heading=h.kppxswuz1tgz)

[9.16 Caso de uso N°26 93](#_heading=h.wvrvop5rh289)

[9.17 Caso de uso N°27 94](#_heading=h.hfmvsov0kmko)

[9.18 Caso de uso N°28 95](#_heading=h.vpkljmhl6s7e)

[9.19 Caso de uso N°29 96](#_heading=h.4774z5za46at)

[9.20 Caso de uso N°30 97](#_heading=h.cgqmkyqzzqfd)

[9.21 Caso de uso N°31 98](#_heading=h.nihs6hvmuf8l)

[9.22 Caso de uso N°32 99](#_heading=h.pz3wcdgzr6ek)

[9.23 Caso de uso N°34 100](#_heading=h.y3c2ccnbuujs)

[9.24 Caso de uso N°38 101](#_heading=h.or4q4sxxuhdx)

[9.25 Caso de uso N°43 102](#_heading=h.hq1vzo348k4c)

[9.26 Caso de uso N°44 103](#_heading=h.367nvnj8vo5h)

[9.27 Caso de uso N°45 104](#_heading=h.xj52hkuba44o)

[9.28 Caso de uso N°46 105](#_heading=h.fndrd8ibua27)

[9.29 Caso de uso N°47 106](#_heading=h.c6i9u9qvkyck)

[9.30 Caso de uso N°48 107](#_heading=h.bbvd92u3a702)

[**10. ÁRBOL DE NAVEGACIÓN 108**](#_heading=h.st9be5x1f8m)

[10.1 Vista “Sistema” 108](#_heading=h.pxrs84bdaoc)

[10.2 Vista “Crear Cuenta” 110](#_heading=h.n1s08pwx1glu)

[10.3 Vista “Recuperar Cuenta” 111](#_heading=h.bc42fe6nz2nd)

[10.5 Vista “Datos Empresa” 113](#_heading=h.7n90vjm33wrs)

[10.6 Vista “Crear Trabajadores” 114](#_heading=h.g9e8dfd1y4ss)

[10.6 Vista “Buscar Trabajador” 114](#_heading=h.46597xuyk0oh)

[10.7 Vista “Ver Trabajador” 115](#_heading=h.rx21f5vz5xpv)

[10.8 Vista “Crear Plantillas de contrato” 115](#_heading=h.d4x3c7n4arle)

[10.9 Vista “Gestión de Descargas” 116](#_heading=h.e0tthdwor7ga)

[10.10 Vista “Cargos” 116](#_heading=h.tr2rmler7zl4)

[10.11 Vista “REC ODI” 117](#_heading=h.442khiwdkj32)

[**11. PRUEBAS 118**](#_heading=h.3c6m06bitixd)

[**12. TABLA BACKLOG CON TIEMPO REAL, TABLA DE ESFUERZO, GRÁFICO Y GRÁFICO BURN-DOWN 151**](#_heading=h.q56tf2k5x8zb)

# 

# **ÍNDICE DE FIGURAS**

[**ÍNDICE DE CONTENIDOS 2**](#_heading=h.k516z6vyokc4)

[**ÍNDICE DE FIGURAS 4**](#_heading=h.2ysda09m44cx)

[**ÍNDICE DE TABLAS 7**](#_heading=h.henznhmwzn5e)

[**1. INTRODUCCIÓN 10**](#_heading=h.qql8w7f98v8b)

[**2. PROPUESTA DE SOLUCIÓN E INCREMENTO 11**](#_heading=h.mmw9v3nn1km5)

[**3. CASOS DE USOS SELECCIONADOS 13**](#_heading=h.5o150tihc5ak)

[**4. TABLA BACKLOG SIN TIEMPO REAL 73**](#_heading=h.f5l715nhceva)

[**5. DIAGRAMA DE CLASES 74**](#_heading=h.pt6ubw1uclfr)

[Figura 5.1 “Diagrama de clases” 74](#_heading=h.asj3m5nxwmut)

[**6. DIAGRAMA DE DATOS 75**](#_heading=h.kpmznybbb2vq)

[Figura 6.1 “Diagrama de datos” 75](#_heading=h.oj34ip1shyc)

[**7. DIAGRAMA DE COMPONENTES 76**](#_heading=h.3s0ik8ssjmy4)

[Figura 7.1 “Diagrama de componentes” 76](#_heading=h.jonbvcl9p0hq)

[**8. DIAGRAMA DE DESPLIEGUE 77**](#_heading=h.v0hhcq3u4bg8)

[Figura 8.1 “Diagrama de despliegue” 77](#_heading=h.4oxddjz8l1h)

[**9. DIAGRAMA DE SECUENCIA 78**](#_heading=h.a76w9c3s6lgn)

[Figura 9.1 “Diagrama de Secuencia - caso de uso 11” 78](#_heading=h.cv05i0xqpd7w)

[Figura 9.2 “Diagrama de Secuencia - caso de uso 12” 79](#_heading=h.meob05nyjk7)

[Figura 9.3 “Diagrama de Secuencia - caso de uso 13” 80](#_heading=h.delrufx5k1w4)

[Figura 9.4 “Diagrama de Secuencia - caso de uso 14” 81](#_heading=h.cxowz6waqoln)

[Figura 9.5 “Diagrama de Secuencia - caso de uso 15” 82](#_heading=h.n06aa09s9mph)

[Figura 9.6 “Diagrama de Secuencia - caso de uso 16” 83](#_heading=h.2axgc6me66kv)

[Figura 9.7 “Diagrama de Secuencia - caso de uso 17” 84](#_heading=h.a5r7eaw2n5eh)

[Figura 9.8 “Diagrama de Secuencia - caso de uso 18” 85](#_heading=h.bfofg6kmy9hg)

[Figura 9.9 “Diagrama de Secuencia - caso de uso 19” 86](#_heading=h.52y48t1gkqum)

[Figura 9.10 “Diagrama de Secuencia - caso de uso 20” 87](#_heading=h.u24pwxfspu5o)

[Figura 9.11 “Diagrama de Secuencia - caso de uso 21” 88](#_heading=h.g7t61eaqkulm)

[Figura 9.12 “Diagrama de Secuencia - caso de uso 22” 89](#_heading=h.cfsghaccoav)

[Figura 9.13 “Diagrama de Secuencia - caso de uso 23” 90](#_heading=h.rg2ahrhv8j2l)

[Figura 9.14 “Diagrama de Secuencia - caso de uso 24” 91](#_heading=h.om2veh9hlrok)

[Figura 9.15 “Diagrama de Secuencia - caso de uso 25” 92](#_heading=h.vefwjmz2ep3u)

[Figura 9.16 “Diagrama de Secuencia - caso de uso 26” 93](#_heading=h.pddk3gx96lxa)

[Figura 9.17 “Diagrama de Secuencia - caso de uso 27” 94](#_heading=h.w6e9lad5w5k7)

[Figura 9.18 “Diagrama de Secuencia - caso de uso 28” 95](#_heading=h.gv64e1a5xqvh)

[Figura 9.19 “Diagrama de Secuencia - caso de uso 29” 96](#_heading=h.3cpfk183du2m)

[Figura 9.20 “Diagrama de Secuencia - caso de uso 30” 97](#_heading=h.mcmh15cktu73)

[Figura 9.21 “Diagrama de Secuencia - caso de uso 31” 98](#_heading=h.4vss84tc0bm4)

[Figura 9.22 “Diagrama de Secuencia - caso de uso 32” 99](#_heading=h.gtcj65lzqfsq)

[Figura 9.23 “Diagrama de Secuencia - caso de uso 34” 100](#_heading=h.uk9g3736ls1c)

[Figura 9.24 “Diagrama de Secuencia - caso de uso 38” 101](#_heading=h.avr1ozh5fx0z)

[Figura 9.25 “Diagrama de Secuencia - caso de uso 43” 102](#_heading=h.lbrx2lwogyah)

[Figura 9.26 “Diagrama de Secuencia - caso de uso 44” 103](#_heading=h.xb6w87d3czri)

[Figura 9.27 “Diagrama de Secuencia - caso de uso 45” 104](#_heading=h.bfajod4dv7b1)

[Figura 9.28 “Diagrama de Secuencia - caso de uso 46” 105](#_heading=h.mf6abp8v0vc0)

[Figura 9.29 “Diagrama de Secuencia - caso de uso 47” 106](#_heading=h.uum2tgsm0m9i)

[Figura 9.30 “Diagrama de Secuencia - caso de uso 48” 107](#_heading=h.t6gayvw4m57y)

[**10. ÁRBOL DE NAVEGACIÓN 108**](#_heading=h.st9be5x1f8m)

[Figura 10.1 “Diagrama de navegación” 108](#_heading=h.xk3zlkkvrxkn)

[Figura 10.2 “Vista “Sistema” 109](#_heading=h.n3myxoxs9d2w)

[Figura 10.3 “Crear Cuenta” 110](#_heading=h.4egw8lg3tbot)

[Figura 10.4 “Recuperar Cuenta 111](#_heading=h.4qp463ri24kv)

[Figura 10.5 “Iniciar Sesión” 112](#_heading=h.tgs2aol3ffdk)

[Figura 10.6 “Datos Empresa” 113](#_heading=h.1fgjyp38hh0a)

[Figura 10.7 “Crear Trabajadores” 114](#_heading=h.gqbfun1frjnv)

[Figura 10.8 “Buscar Trabajador” 114](#_heading=h.v6euucradsq9)

[Figura 10.8 “Buscar Trabajador” 115](#_heading=h.6xizgo1h4z7i)

[Figura 10.9 “Buscar Trabajador” 115](#_heading=h.tagkjvhrkdcr)

[Figura 10.10 “Gestión de descargas” 116](#_heading=h.79kx5vm335cx)

[Figura 10.11 “Gestión de descargas” 116](#_heading=h.4xc60wljfyvc)

[Figura 10.12 “REC ODI” 117](#_heading=h.hu9gv666fao)

[**11. PRUEBAS 118**](#_heading=h.3c6m06bitixd)

[Figura 11.1 “Pruebas del sistema - Caso de uso 11” 119](#_heading=h.jzl1c3sxxeo7)

[Figura 11.1 “Pruebas del sistema - Caso de uso 11” 119](#_heading=h.nk3utmf3hnvy)

[Figura 11.1 “Pruebas del sistema - Caso de uso 11” 120](#_heading=h.v346cvcbk3rt)

[Figura 11.2 “Pruebas del sistema” 121](#_heading=h.s2non8g92nl4)

[Figura 11.3 “Pruebas del sistema” 122](#_heading=h.wpmhaypf4168)

[Figura 11.4 “Pruebas del sistema” 123](#_heading=h.1lj5f4fqpy8e)

[Figura 11.5 “Pruebas del sistema” 124](#_heading=h.n47hsv72afv6)

[Figura 11.6 “Pruebas del sistema” 124](#_heading=h.cpu96vs9ldom)

[Figura 11.7 “Pruebas del sistema” 125](#_heading=h.f6deecvbuvyf)

[Figura 11.8 “Pruebas del sistema” 126](#_heading=h.6ctjkml7ynh1)

[Figura 11.9 “Pruebas del sistema” 127](#_heading=h.y9jdhlxf80rc)

[Figura 11.10 “Pruebas del sistema” 128](#_heading=h.4ltgmcm3wtvi)

[Figura 11.11 “Pruebas del sistema” 128](#_heading=h.ajekuyhq5wiv)

[Figura 11.12 “Pruebas del sistema” 129](#_heading=h.thwrz6iz47kb)

[Figura 11.13 “Pruebas del sistema” 130](#_heading=h.gxd6qzw9mtmb)

[Figura 11.14 “Pruebas del sistema” 130](#_heading=h.l0ov2wg1l4xr)

[Figura 11.15 “Pruebas del sistema” 131](#_heading=h.ancanps2f7qz)

[Figura 11.13 “Pruebas del sistema” 132](#_heading=h.spnd3cyqfflp)

[Figura 11.14 “Pruebas del sistema” 133](#_heading=h.fp3hxzz3w4ts)

[Figura 11.15 “Pruebas del sistema” 134](#_heading=h.vcbvfkaskzld)

[Figura 11.16 “Pruebas del sistema” 135](#_heading=h.455g5uerziva)

[Figura 11.17 “Pruebas del sistema” 136](#_heading=h.cwlj26nw5s9j)

[Figura 11.18 “Pruebas del sistema” 137](#_heading=h.a121kr1hv0ap)

[Figura 11.19 “Pruebas del sistema” 138](#_heading=h.h2bugwxr6xle)

[Figura 11.20 “Pruebas del sistema” 138](#_heading=h.okm7hvn5kmxp)

[Figura 11.21 “Pruebas del sistema” 139](#_heading=h.3e43nfmz0zok)

[Figura 11.22 “Pruebas del sistema” 139](#_heading=h.9y2mm52si0tn)

[Figura 11.23 “Pruebas del sistema” 140](#_heading=h.jrld8a8ov3lv)

[Figura 11.24 “Pruebas del sistema” 141](#_heading=h.tftexm1ex7ms)

[Figura 11.25 “Pruebas del sistema” 142](#_heading=h.iw73gen43dyc)

[Figura 11.26 “Pruebas del sistema” 142](#_heading=h.n7yqpwvk77dm)

[Figura 11.27 “Pruebas del sistema” 143](#_heading=h.7j3qp9ee9rsn)

[Figura 11.28 “Pruebas del sistema” 143](#_heading=h.hgc32q1jrbsy)

[Figura 11.29 “Pruebas del sistema” 144](#_heading=h.yfvwgvg25x2j)

[Figura 11.30 “Pruebas del sistema” 145](#_heading=h.bxp2cdukix6v)

[Figura 11.31 “Pruebas del sistema” 146](#_heading=h.5cu7eqva3myv)

[Figura 11.32 “Pruebas del sistema” 147](#_heading=h.c48lnydo6ntv)

[Figura 11.33 “Pruebas del sistema” 148](#_heading=h.n70rdnbh5dia)

[Figura 11.34 “Pruebas del sistema” 149](#_heading=h.n4ac44sj2ej6)

[Figura 11.35 “Pruebas del sistema” 150](#_heading=h.e5xjoguz5azo)

[**12. TABLA BACKLOG CON TIEMPO REAL, TABLA DE ESFUERZO, GRÁFICO Y GRÁFICO BURN-DOWN 151**](#_heading=h.q56tf2k5x8zb)

[Figura 12.3 – “Diagrama de burn-down” 152](#_heading=h.82287a1zcvdn)

[Figura 12.4 – “Diagrama de burn-down” 153](#_heading=h.vkch8hux1dzz)

# **ÍNDICE DE TABLAS**

[**ÍNDICE DE CONTENIDOS 2**](#_heading=h.k516z6vyokc4)

[**ÍNDICE DE FIGURAS 4**](#_heading=h.2ysda09m44cx)

[**ÍNDICE DE TABLAS 7**](#_heading=h.henznhmwzn5e)

[**1. INTRODUCCIÓN 10**](#_heading=h.qql8w7f98v8b)

[**2. PROPUESTA DE SOLUCIÓN E INCREMENTO 11**](#_heading=h.mmw9v3nn1km5)

[Tabla 2.1 – “Tabla solución e incremento” 12](#_heading=h.d9t9t43qas6x)

[**3. CASOS DE USOS SELECCIONADOS 13**](#_heading=h.5o150tihc5ak)

[Tabla 3.1: “Caso de uso N°11” 13](#_heading=h.b12cvbv5xi8o)

[Tabla 3.2: “Caso de uso N°12” 15](#_heading=h.poc56rkfi8at)

[Tabla 3.3: “Caso de uso N°13” 17](#_heading=h.58ekpj9u4qvo)

[Tabla 3.4: “Caso de uso N°14” 19](#_heading=h.luge9ejaeoc4)

[Tabla 3.5: “Caso de uso N°15” 21](#_heading=h.1r1kawypuzc3)

[Tabla 3.6: “Caso de uso N°16” 23](#_heading=h.gh4r6bhjset2)

[Tabla 3.7: “Caso de uso N°17” 25](#_heading=h.dvncke2jlc56)

[Tabla 3.8: “Caso de uso N°18” 27](#_heading=h.xhlri6y9c2ny)

[Tabla 3.9: “Caso de uso N°19” 29](#_heading=h.6ndw6wy9n22i)

[Tabla 3.10: “Caso de uso N°20” 31](#_heading=h.v0e0xlsytn1d)

[Tabla 3.11: “Caso de uso N°21” 33](#_heading=h.8dofvl7nya79)

[Tabla 3.12: “Caso de uso N°22” 35](#_heading=h.wtu9olt0ne33)

[Tabla 3.13: “Caso de uso N°23” 37](#_heading=h.on2m0lqqfjbm)

[Tabla 3.14: “Caso de uso N°24” 39](#_heading=h.iv0ycgi6gq7v)

[Tabla 6.15: “Caso de uso N°25” 41](#_heading=h.1zubfnj2e8fz)

[Tabla 6.16: “Caso de uso N°26” 43](#_heading=h.8kgp2ljsgv1s)

[Tabla 6.17: “Caso de uso N°27” 45](#_heading=h.cqpv35kyq4nz)

[Tabla 3.18: “Caso de uso N°28” 47](#_heading=h.s9hfeexaqge4)

[Tabla 3.19: “Caso de uso N°29” 49](#_heading=h.pmrso0zckk4c)

[Tabla 3.20: “Caso de uso N°30” 51](#_heading=h.2an0wkgekjyk)

[Tabla 3.21: “Caso de uso N°31” 53](#_heading=h.yua6zlwcucnb)

[Tabla 3.22: “Caso de uso N°32” 55](#_heading=h.kugupsfksubu)

[Tabla 3.23: “Caso de uso N°34” 57](#_heading=h.ny5ybi5t0uvx)

[Tabla 3.24: “Caso de uso N°38” 59](#_heading=h.p9zz91a4mjc9)

[Tabla 3.25: “Caso de uso N°43” 61](#_heading=h.e999o4o6bkpd)

[Tabla 3.26: “Caso de uso N°44” 63](#_heading=h.jim9d3ogkepr)

[Tabla 3.27: “Caso de uso N°45” 65](#_heading=h.uxr19uoprwdf)

[Tabla 3.28: “Caso de uso N°46” 67](#_heading=h.oedqvoqk02lg)

[Tabla 3.29: “Caso de uso N°47” 69](#_heading=h.yqff0ytjmtj2)

[Tabla 3.30: “Caso de uso N°48” 71](#_heading=h.sub0xystgwvv)

[**4. TABLA BACKLOG SIN TIEMPO REAL 73**](#_heading=h.f5l715nhceva)

[Tabla 4.1 – “Tabla backlog” 73](#_heading=h.azwzxc76mgmr)

[**5. DIAGRAMA DE CLASES 74**](#_heading=h.pt6ubw1uclfr)

[**6. DIAGRAMA DE DATOS 75**](#_heading=h.kpmznybbb2vq)

[**7. DIAGRAMA DE COMPONENTES 76**](#_heading=h.3s0ik8ssjmy4)

[**8. DIAGRAMA DE DESPLIEGUE 77**](#_heading=h.v0hhcq3u4bg8)

[**9. DIAGRAMA DE SECUENCIA 78**](#_heading=h.a76w9c3s6lgn)

[9.1 Caso de uso N°11 78](#_heading=h.kv8w5h2um89r)

[9.2 Caso de uso N°12 79](#_heading=h.qgfxkonr3mxd)

[9.3 Caso de uso N°13 80](#_heading=h.4fv8x4lfyefl)

[9.4 Caso de uso N°14 81](#_heading=h.ft06c4n6hb5c)

[9.5 Caso de uso N°15 82](#_heading=h.d98yu5kr1zqg)

[9.6 Caso de uso N°16 83](#_heading=h.of0sdir2d2u2)

[9.7 Caso de uso N°17 84](#_heading=h.3kjnzx89h30u)

[9.8 Caso de uso N°18 85](#_heading=h.q6vd28ym2zzx)

[9.9 Caso de uso N°19 86](#_heading=h.ij96edrtr81)

[9.10 Caso de uso N°20 87](#_heading=h.2cpalrkx9fjp)

[9.11 Caso de uso N°21 88](#_heading=h.u23g0vdk4afd)

[9.12 Caso de uso N°22 89](#_heading=h.jnf7ztmigk24)

[9.13 Caso de uso N°23 90](#_heading=h.2srubtxa4bb6)

[9.14 Caso de uso N°24 91](#_heading=h.dundakiubkma)

[9.15 Caso de uso N°25 92](#_heading=h.kppxswuz1tgz)

[9.16 Caso de uso N°26 93](#_heading=h.wvrvop5rh289)

[9.17 Caso de uso N°27 94](#_heading=h.hfmvsov0kmko)

[9.18 Caso de uso N°28 95](#_heading=h.vpkljmhl6s7e)

[9.19 Caso de uso N°29 96](#_heading=h.4774z5za46at)

[9.20 Caso de uso N°30 97](#_heading=h.cgqmkyqzzqfd)

[9.21 Caso de uso N°31 98](#_heading=h.nihs6hvmuf8l)

[9.22 Caso de uso N°32 99](#_heading=h.pz3wcdgzr6ek)

[9.23 Caso de uso N°34 100](#_heading=h.y3c2ccnbuujs)

[9.24 Caso de uso N°38 101](#_heading=h.or4q4sxxuhdx)

[9.25 Caso de uso N°43 102](#_heading=h.hq1vzo348k4c)

[9.26 Caso de uso N°44 103](#_heading=h.367nvnj8vo5h)

[9.27 Caso de uso N°45 104](#_heading=h.xj52hkuba44o)

[9.28 Caso de uso N°46 105](#_heading=h.fndrd8ibua27)

[9.29 Caso de uso N°47 106](#_heading=h.c6i9u9qvkyck)

[9.30 Caso de uso N°48 107](#_heading=h.bbvd92u3a702)

[**10. ÁRBOL DE NAVEGACIÓN 108**](#_heading=h.st9be5x1f8m)

[10.1 Vista “Sistema” 108](#_heading=h.pxrs84bdaoc)

[10.2 Vista “Crear Cuenta” 110](#_heading=h.n1s08pwx1glu)

[10.3 Vista “Recuperar Cuenta” 111](#_heading=h.bc42fe6nz2nd)

[10.5 Vista “Datos Empresa” 113](#_heading=h.7n90vjm33wrs)

[10.6 Vista “Crear Trabajadores” 114](#_heading=h.g9e8dfd1y4ss)

[10.6 Vista “Buscar Trabajador” 114](#_heading=h.46597xuyk0oh)

[10.7 Vista “Ver Trabajador” 115](#_heading=h.rx21f5vz5xpv)

[10.8 Vista “Crear Plantillas de contrato” 115](#_heading=h.d4x3c7n4arle)

[10.9 Vista “Gestión de Descargas” 116](#_heading=h.e0tthdwor7ga)

[10.10 Vista “Cargos” 116](#_heading=h.tr2rmler7zl4)

[10.11 Vista “REC ODI” 117](#_heading=h.442khiwdkj32)

[**11. PRUEBAS 118**](#_heading=h.3c6m06bitixd)

[**12. TABLA BACKLOG CON TIEMPO REAL, TABLA DE ESFUERZO, GRÁFICO Y GRÁFICO BURN-DOWN 151**](#_heading=h.q56tf2k5x8zb)

[Tabla 12.1 – “Tabla backlog con tiempo real” 151](#_heading=h.ls5i39y5ctz)

[Tabla 12.2 – “Tabla de esfuerzo” 152](#_heading=h.omgwxdj3pzjt)

### 

# 

# 1. INTRODUCCIÓN

Este documento presenta la aplicación de la metodología Scrum++, un enfoque híbrido que combina los principios iterativos de Scrum con la perspectiva estructural del modelo arquitectónico 4+1, con el objetivo de gestionar y dar trazabilidad al desarrollo de un sistema ERP web. En el anexo disponible en el Drive se incluye además el informe generado a partir de GitHub, junto con el log completo de actividad.

Scrum ++ permite organizar el trabajo de manera incremental, enfocándose tanto en la entrega de valor funcional como en la definición técnica del sistema desde múltiples vistas arquitectónicas. Este enfoque facilita una planificación más clara, un seguimiento continuo del avance y una documentación coherente con los componentes implementados.

A lo largo del documento se presentan los principales artefactos generados durante la ejecución del Sprint, incluyendo:

* Propuesta de solución e incremento desarrollado
* Casos de uso seleccionados
* Tabla del backlog estimado (sin tiempos reales)
* Diagrama de clases
* Diagrama de datos
* Diagrama de componentes
* Diagrama de despliegue
* Diagrama de secuencia y árbol de navegación
* Pruebas realizadas
* Tabla del backlog real con tiempos y esfuerzo registrado
* Tabla de esfuerzo distribuido por tarea y fecha
* Gráfico de esfuerzo restante (Burn-down Chart)

Cada uno de estos elementos contribuye a la trazabilidad del proyecto, permitiendo evidenciar el cumplimiento de los objetivos planteados, el seguimiento del trabajo planificado y la correcta integración de los componentes del sistema. Esta documentación respalda la gestión ágil del desarrollo, asegurando la calidad técnica y el control del progreso en cada fase del proceso.

# 2. PROPUESTA DE SOLUCIÓN E INCREMENTO

La planificación del desarrollo del sistema ERP ContaPlus se ha organizado en incrementos, cada uno representando una fase del proyecto compuesta por una cantidad determinada de casos de uso. Esta estrategia permite abordar el desarrollo de manera modular, facilitando la validación progresiva de funcionalidades, la retroalimentación oportuna y la reducción de errores acumulativos.

##### Tabla 2.1 – “Tabla solución e incremento”

| Incremento | Casos de Uso | Porcentaje Parcial | Porcentaje Acumulado |
| --- | --- | --- | --- |
| Incremento 1 | 10 | 20% | 20% |
| Incremento 2 | 30% | 60% | 80% |

# 3. CASOS DE USOS SELECCIONADOS

En esta sección se presentan los casos de uso seleccionados para el Avance Scrum++ 2, los cuales constituyen los cimientos esenciales del sistema ContaPlus. Cada uno describe la interacción entre los actores y la aplicación, especificando sus objetivos, las precondiciones necesarias y el flujo principal de eventos. El propósito es formalizar las funcionalidades críticas de esta primera iteración, garantizando que estén alineadas con los requerimientos del usuario y sirvan de guía para el diseño y la validación del sistema.

A continuación desde la Tabla 3.1 hasta la Tabla 3.30 se presentan los casos de uso del Scrum ++ 2

##### *Tabla 3.1: “Caso de uso N°11”*

| caso de uso N°11 | agregando método de pago a socio |
| --- | --- |
| Actores | Administrador |
| Propósito | Registrar una nueva forma de pago asociada a un socio para gestionar depósitos o transferencias. |
| Precondiciones | El socio debe estar registrado y su perfil debe estar accesible. |
| Resumen | El sistema permite completar un formulario con los datos bancarios del socio para añadir un nuevo método de pago. |
| Postcondiciones | El nuevo método de pago queda registrado y disponible en el perfil del socio. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al perfil del socio | 2. muestra sección de métodos de pago |
| 3. presiona “agregar nuevo método de pago” | 4. despliega el formulario de ingreso |
| 5. completa los campos y confirma | 6. guarda el método de pago y muestra mensaje de éxito |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.2: “Caso de uso N°12”*

| caso de uso N°12 | editando método de pago de socio |
| --- | --- |
| Actores | Administrador |
| Propósito | Modificar los datos o condiciones de un método de pago previamente registrado para un socio. |
| Precondiciones | El socio debe estar registrado con al menos un método de pago asociado. |
| Resumen | El sistema permite editar el tipo de cuenta, banco, titular u otros datos del método de pago seleccionado. |
| Postcondiciones | El método de pago queda actualizado y activo para futuras operaciones. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al perfil del socio | 2. muestra la lista de métodos de pago existentes |
| 3. selecciona el método que desea modificar | 4. despliega los campos editables con los datos actuales |
| 5. actualiza los campos y confirma | 6. guarda los cambios y confirmó la edición |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### 

# 

# 

# 

# 

# 

##### *Tabla 3.3: “Caso de uso N°13”*

| caso de uso N°13 | eliminando método de pago de socio |
| --- | --- |
| Actores | Administrador |
| Propósito | Eliminar una forma de pago previamente asignada a un socio registrado. |
| Precondiciones | El socio debe estar registrado y tener al menos un método de pago asignado. |
| Resumen | El sistema permite seleccionar un método de pago específico desde el perfil del socio y eliminarlo. |
| Postcondiciones | El método de pago es eliminado y ya no se encuentra disponible para operaciones futuras. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al perfil del socio | 2. muestra la lista de métodos de pago registrados |
| 3. selecciona el método de pago a eliminar | 4. muestra confirmación de eliminación |
| 5. confirma la eliminación | 6. elimina el método y actualiza la vista del socio |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

.

##### *Tabla 3.4: “Caso de uso N°14”*

| caso de uso N°14 | editando acciones y capital total de la empresa |
| --- | --- |
| Actores | Administrador |
| Propósito | Modificar la estructura del capital total y la distribución de acciones en la empresa. |
| Precondiciones | Debe existir al menos un socio registrado y el administrador debe tener permisos de edición. |
| Resumen | El sistema permite cambiar el capital total, valor de la acción o porcentajes asociados a los socios. |
| Postcondiciones | Los nuevos valores quedan actualizados y disponibles para revisión o cálculos posteriores. |
| Tipos | Principal – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede a la sección de acciones y capital | 2. muestra datos actuales de capital y socios |
| 3. edita los valores correspondientes | 4. válida consistencia entre valores, acciones y porcentajes |
| 5. guarda la información | 6. actualiza los valores y confirma cambios exitosos |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.5: “Caso de uso N°15”*

| caso de uso N°15 | editando direcciones de trabajo de la empresa |
| --- | --- |
| Actores | Administrador |
| Propósito | Modificar las direcciones de las distintas sucursales o lugares de trabajo asociados a la empresa. |
| Precondiciones | La empresa debe estar registrada y debe tener direcciones previamente cargadas. |
| Resumen | El sistema permite acceder a la sección de direcciones laborales, editar los campos existentes como calle, número y comuna, y guardar los cambios realizados. |
| Postcondiciones | Las nuevas direcciones quedan registradas en el sistema y disponibles para su uso en contratos u otros procesos administrativos. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede al módulo de direcciones laborales | 2. muestra la lista de direcciones existentes |
| 3. selecciona la dirección que desea editar | 4. despliega el formulario editable con los datos actuales |
| 5. modifica los campos y guarda los cambios | 6. actualiza la dirección y muestra mensaje de confirmación |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.6: “Caso de uso N°16”*

| caso de uso N°16 | editando información de seguridad y previsión de la empresa |
| --- | --- |
| Actores | Administrador de empresa |
| Propósito | Actualizar la información relacionada con mutualidades, cajas de compensación, entidades de salud y previsión de la empresa. |
| Precondiciones | La empresa debe estar registrada y el usuario debe tener permisos para editar estos datos. |
| Resumen | El sistema permite acceder a la sección de seguridad y previsión, editar los campos existentes como mutualidad, caja de compensación y AFP, y guardar los cambios. |
| Postcondiciones | La información actualizada queda disponible para su uso en la gestión de trabajadores y contratos. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede al módulo de seguridad y previsión | 2. muestra los datos actuales registrados |
| 3. selecciona los campos que desea modificar | 4. habilita los campos para edición |
| 5. realiza los cambios y presiona “guardar cambios” | 6. actualiza la información y muestra mensaje de confirmación |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.7: “Caso de uso N°17”*

| caso de uso N°17 | editando actividad económica y régimen tributario de la empresa |
| --- | --- |
| Actores | Administrador |
| Propósito | Actualizar los datos de la actividad económica principal y el régimen tributario bajo el cual opera la empresa. |
| Precondiciones | La empresa debe estar registrada y el usuario debe tener permisos de edición sobre estos datos. |
| Resumen | El sistema permite modificar la actividad económica principal (giro), rubro o sector, así como el régimen tributario vigente, seleccionando entre opciones. |
| Postcondiciones | Los nuevos datos quedan registrados en el sistema y se consideran en documentos legales y formularios. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede al módulo de actividad económica y tributos | 2. muestra los datos actuales registrados |
| 3. selecciona los campos a modificar | 4. habilita campos de edición con opciones disponibles |
| 5. realiza cambios y guarda | 6. actualiza los datos |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.8: “Caso de uso N°18”*

| caso de uso N°18 | editando datos legales de la empresa |
| --- | --- |
| Actores | Administrador |
| Propósito | Modificar la información legal de la empresa, como razón social, tipo de sociedad. |
| Precondiciones | La empresa debe estar registrada y el usuario debe contar con permisos de edición. |
| Resumen | El sistema permite acceder a la sección de datos legales de la empresa, editar campos como razón social, notaría, fecha y número de escritura. |
| Postcondiciones | La nueva información legal queda registrada y se utiliza en contratos y documentos oficiales. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede al módulo de datos legales | 2. muestra los datos actuales en campos editables |
| 3. modifica los datos requeridos | 4. valida que los campos cumplan con el formato correspondiente |
| 5. guarda los cambios | 6. actualiza la información y muestra mensaje de confirmación |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.9: “Caso de uso N°19”*

| caso de uso N°19 | editando datos generales de la empresa |
| --- | --- |
| Actores | Administrador |
| Propósito | Modificar la información general de la empresa, como nombre de fantasía, teléfono, correo o sitio web. |
| Precondiciones | La empresa debe estar registrada en el sistema y el usuario debe tener permisos para editar su información. |
| Resumen | El sistema permite ingresar a la sección de información general y editar campos básicos como nombre, contacto y presencia digital de la empresa. |
| Postcondiciones | Los datos generales quedan actualizados en la ficha de empresa y visibles para otros módulos relacionados. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede a la sección de datos generales | 2. muestra los campos actuales en formato editable |
| 3. modifica los datos requeridos | 4. valida formato de correo, teléfono |
| 5. guarda los cambios | 6. actualiza la información y notifica edición exitosa |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.10: “Caso de uso N°20”*

| caso de uso N°20 | creando perfil trabajador |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Registrar todos los datos personales para crear el perfil del trabajador. |
| Precondiciones | El usuario debe estar autenticado con rol de Administrador o Recursos humanos y no debe existir el trabajador registrado. |
| Resumen | El sistema permite completar un formulario dividido en secciones con datos personales. |
| Postcondiciones | El sistema crea un nuevo registro de trabajadores y lo deja disponible para su gestión posterior. |
| Tipos | Principal – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al módulo de registro de trabajadores | 2. muestra el formulario dividido en secciones |
| 3. completa todos los campos requeridos del formulario | 4. valida los datos ingresados |
| 5. presiona el botón “guardar trabajador” | 6. crea el perfil |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.11: “Caso de uso N°21”*

| caso de uso N°21 | Ingresando datos de contrato |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permite registrar en el sistema los datos del contrato (sueldo base, forma de pago, fecha de contrato, jefe, cargo, fecha de ingreso) |
| Precondiciones | El usuario debe estar registrado como administrador o recursos humanos. |
| Resumen | El sistema permite ingresar de manera manual los datos del contrato. |
| Postcondiciones | Los datos quedan guardados para su posterior gestion |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede a la sección de “Crear Trabajadores” | 2. muestra el formulario con los datos necesarios |
| 3. ingresa los datos | 4. verifica formato |
| 5. presiona en “guardar” | 6. guarda los datos y lo asocia al trabajador correspondiente |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.12: “Caso de uso N°22”*

| caso de uso N°22 | Ingresando datos previsionales del trabajador |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Ingresar y guardar los datos previsionales del trabajador (AFP, Isapre/Fonasa) |
| Precondiciones | El usuario debe estar registrado como Administrador o Recursos humanos |
| Resumen | El sistema permite ingresar de manera manual datos previsionales como lo son AFP, isapre/Fonasa, Plan de isapre en uf |
| Postcondiciones | Los datos quedan guardados para su posterior gestión |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede a la sección de “Crear Trabajadores” | 2. muestra el formulario con los datos necesarios |
| 3. Ingresa los datos seleccionados | 4. verifica formato y validez |
| 5. presiona “Guardar” | 6. guarda los datos |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.13: “Caso de uso N°23”*

| caso de uso N°23 | subiendo documentos personales del trabajador |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Adjuntar archivos como contratos, certificados u otros documentos relevantes al perfil del trabajador. |
| Precondiciones | El perfil del trabajador debe estar creado y accesible. |
| Resumen | El sistema permite cargar documentos seleccionados por el usuario desde su dispositivo, vinculados al perfil del trabajador. |
| Postcondiciones | El archivo queda registrado y visible en la sección de documentos personales del trabajador. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede a la sección de documentos del trabajador | 2. muestra los archivos ya cargados y opción para subir uno nuevo |
| 3. selecciona archivo desde su dispositivo | 4. verifica formato y tamaño |
| 5. presiona “subir archivo” | 6. guarda el archivo y lo asocia al trabajador correspondiente |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.14: “Caso de uso N°24”*

| caso de uso N°24 | Editando perfil completo del trabajador |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Editar todos los datos personales, laborales y contractuales. |
| Precondiciones | El usuario debe estar autenticado con rol de Administrador o Recursos humanos y debe existir el trabajador registrado. |
| Resumen | El sistema permite completar un formulario dividido en secciones con datos personales, previsionales, laborales y contractuales del trabajador. |
| Postcondiciones | El sistema actualiza un registro de trabajadores y lo deja disponible para su gestión posterior. |
| Tipos | Principal – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al módulo de ver trabajador | 2. muestra el formulario dividido en secciones |
| 3. completa todos los campos requeridos del formulario | 4. valida los datos ingresados |
| 5. presiona el botón “guardar trabajador” | 6. actualiza el perfil |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 6.15: “Caso de uso N°25”*

| caso de uso N°25 | buscando trabajador por RUT |
| --- | --- |
| Actores | Administrador, Recursos humanos |
| Propósito | Localizar un trabajador específico mediante su número de RUT. |
| Precondiciones | Deben existir trabajadores registrados en el sistema. |
| Resumen | El sistema permite filtrar la tabla de trabajadores ingresando un RUT en el campo de búsqueda. |
| Postcondiciones | Se muestra el trabajador coincidente con el RUT ingresado. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al listado de trabajadores | 2. muestra el listado completo |
| 3. escribe el RUT del trabajador en el buscador | 4. filtra la lista según el RUT ingresado |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 6.16: “Caso de uso N°26”*

| caso de uso N°26 | buscando trabajador por cargo |
| --- | --- |
| Actores | Administrador, Recursos humanos, Contador |
| Propósito | Encontrar uno o más trabajadores mediante su nombre o parte de él. |
| Precondiciones | Deben existir trabajadores registrados en el sistema. |
| Resumen | El sistema filtra dinámicamente la lista de trabajadores según el nombre ingresado. |
| Postcondiciones | Se listan los trabajadores que coinciden parcial o totalmente con el nombre buscado. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al listado de trabajadores | 2. muestra la tabla con todos los registros |
| 3. ingresa el cargo | 4. muestra todos los trabajadores con ese cargo |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 6.17: “Caso de uso N°27”*

| caso de uso N°27 | buscando trabajador por nombre |
| --- | --- |
| Actores | Administrador, Recursos humanos, Contador |
| Propósito | Encontrar uno o más trabajadores mediante su nombre o parte de él. |
| Precondiciones | Deben existir trabajadores registrados en el sistema. |
| Resumen | El sistema filtra dinámicamente la lista de trabajadores según el nombre ingresado. |
| Postcondiciones | Se listan los trabajadores que coinciden parcial o totalmente con el nombre buscado. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al listado de trabajadores | 2. muestra la tabla con todos los registros |
| 3. ingresa el nombre o parte del nombre | 4. Muestra todas las opciones con ese nombre |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.18: “Caso de uso N°28”*

| caso de uso N°28 | visualizando perfil del trabajador |
| --- | --- |
| Actores | Administrador, Recursos humanos, Contador |
| Propósito | Poder ver los datos de un trabajador |
| Precondiciones | Deben existir trabajadores registrados en el sistema, además de que el trabajador debe ser buscado. |
| Resumen | El sistema filtra dinámicamente la lista de trabajadores según el nombre ingresado. |
| Postcondiciones | El sistema muestra el perfil completo del trabajador |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede a la sección de “Buscar Trabajador” | 2. lista los trabajadores |
| 3. Busca trabajador por rut, nombre o cargo | 4. Muestra los trabajadores filtrados |
| 5. selecciona al trabajador buscado | 6. muestra su información en pantalla |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.19: “Caso de uso N°29”*

| caso de uso N°29 | Editando datos previsionales |
| --- | --- |
| Actores | Administrador, Recursos humanos |
| Propósito | Poder editar datos como afp, fonasa/isapre, valor de la isapre |
| Precondiciones | Deben existir trabajadores registrados en el sistema, además de que el trabajador debe ser buscado. |
| Resumen | El sistema permite editar los datos previsionales de un trabajador en específico |
| Postcondiciones | El sistema actualiza el perfil completo del trabajador |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede a la sección de “Buscar Trabajador” | 2. lista los trabajadores |
| 3. Busca trabajador por rut, nombre o cargo | 4. Muestra los trabajadores filtrados |
| 5. selecciona al trabajador buscado | 6. muestra su información en pantalla |
| 7. edita la información previsional y selecciona en guardar | 8. Actualiza el perfil del trabajador |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.20: “Caso de uso N°30”*

| caso de uso N°30 | creando cargo |
| --- | --- |
| Actores | Administrador, Recursos humanos |
| Propósito | Diseñar una plantilla estandarizada para describir cargos dentro de la empresa. |
| Precondiciones | El administrador debe tener acceso a la sección de plantillas y permisos para crear una nueva. |
| Resumen | El sistema permite ingresar el título del cargo, y una descripción libre |
| Postcondiciones | La nueva plantilla queda guardada y disponible para asignarse a trabajadores o descargarse. |
| Tipos | Principal – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede al módulo de plantillas de cargo | 2. muestra opción para “crear nueva plantilla” |
| 3. ingresa título y su descripción | 4. valida campos obligatorios |
| 5. presiona “guardar” | 6. registra la plantilla |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.21: “Caso de uso N°31”*

| caso de uso N°31 | Buscando cargo por nombre |
| --- | --- |
| Actores | Administrador, Recursos humanos |
| Propósito | Permitir buscar un cargo por su nombre |
| Precondiciones | El administrador debe tener acceso a la sección de plantillas y deben existir cargos |
| Resumen | El sistema permite buscar un cargo mediante su nombre o partes de su nombre |
| Postcondiciones | El cargo queda listo para ser visualizado o descargado |
| Tipos | Principal – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. accede al módulo de plantillas de cargo | 2. muestra la vista de cargos” |
| 3. ingresa un nombre o parte de uno | 4. devuelve todas las coincidencias |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.22: “Caso de uso N°32”*

| caso de uso N°32 | descargando descripción de cargo en formato .PDF |
| --- | --- |
| Actores | Administrador, Recursos humanos |
| Propósito | Obtener una copia descargable en formato .PDF de una plantilla de descripción de cargo previamente creada o editada. |
| Precondiciones | Debe existir una plantilla de descripción de cargo registrada en el sistema. |
| Resumen | El sistema permite seleccionar una plantilla de descripción de cargo, generar su versión en PDF con el formato oficial y descargarla para ser utilizada en procesos de contratación o archivo administrativo. |
| Postcondiciones | La plantilla queda descargada en el dispositivo del usuario en formato .PDF y lista para su impresión o uso. |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo de descripciones de cargo | 2. Muestra la lista de plantillas disponibles |
| 3. Selecciona una plantilla | 4. Genera la vista previa en formato imprimible |
| 5. Presiona el botón “descargar en PDF” | 6. Genera el archivo y activa la descarga en el navegador |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.23: “Caso de uso N°34”*

| caso de uso N°34 | Visualizando plantilla de contrato |
| --- | --- |
| Actores | Administrador, Recursos humanos |
| Propósito | Permite ver una plantilla de contrato con sus variables dinámicas. |
| Precondiciones | El usuario debe tener rol de Administrador o Recursos humanos y debe existir una plantilla |
| Resumen | El sistema permite visualizar una plantilla para poder administrar o buscar plantillas en especifico |
| Postcondiciones | La plantilla queda cerrada y no se modifica nada de ella |
| Tipos | Secundario – Esencial – Técnico |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo “Ver plantillas” | 2. Muestra las plantillas disponibles |
| 3. Selecciona la plantilla que requiera | 4. Muestra la plantilla |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.24: “Caso de uso N°38”*

| caso de uso N°38 | Agregando anexo de contrato |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permite agregar un anexo a un contrato existente |
| Precondiciones | El usuario debe estar autenticado con el rol de Recursos Humanos o Administrador y tener acceso al módulo de Anexos de contrato. |
| Resumen | El sistema permite crear un anexo de contrato y luego poder adjuntarlo a contratos ya existentes |
| Postcondiciones | El anexo queda adjuntado al contrato seleccionado |
| Tipos | Principal – Esencial – Técnico |
| Curso normal de eventos | |
| acción de los actores | respuesta del sistema |
| 1. accede al módulo de anexos contractuales | 2. muestra la opción “crear nuevo anexo” |
| 3. selecciona la opción de creación | 4. despliega editor con texto libre para ingresar el anexo |
| 5. completa la información requerida | 6. guarda la plantilla y muestra mensaje de confirmación |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.25: “Caso de uso N°43”*

| Caso de Uso N°43 | Creando O.D.I |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permitir crear un O.D.I |
| Precondiciones | El usuario debe estar registrado anteriormente. |
| Resumen | El sistema permite crear una O.D.I con sus datos: Actividad, Peligros, Consecuencias, Medidas de control y los datos del trabajador |
| Postcondiciones | El O.D.I queda registrado para posteriormente poder ser descargado |
| Tipos | Principal – Documental – Financiero |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo de “REC ODI” | 1. Muestra la vista del módulo |
| 1. Rellena los datos del trabajador | 1. Renderiza la previsualización |
| 1. Clic en “Crear ODI” | 1. muestra la vista de “Cear ODI” |
| 1. Rellena los datos solicitados y presiona en guardar | 1. Muestra los datos en la vista |
| 1. Clic en Descargar PDF | 1. descarga el pdf, además de guardarlo en el sistema |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.26: “Caso de uso N°44”*

| Caso de Uso N°44 | Subiendo pdf de O.D.I a perfil del trabajador |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permitir subir el archivo pdf firmado de los O.D.I entregados |
| Precondiciones | El usuario debe estar registrado anteriormente, |
| Resumen | El sistema permite subir los pdf generador en asignando O.D.Ial perfil del trabajador |
| Postcondiciones | El O.D.I queda registrado en el perfil del trabajador para poder ser descargado |
| Tipos | Principal – Documental – Financiero |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo de “Ver Trabajador” | 1. Muestra la vista del módulo |
| 1. Buscar trabajador por rut o nombre o cargo | 1. Lista los trabajadores en base a la petición |
| 1. Sube el archivo en la zona de “Registro de O.D.I” y hace clic en “Actualizar” | 1. Valida tipo de archivo y lo guarda en el sistema |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.27: “Caso de uso N°45”*

| Caso de Uso N°45 | Descargando pdf de O.D.I del perfil del trabajador |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permitir descargar el archivo pdf firmado de los O.D.I entregados |
| Precondiciones | El usuario debe estar registrado anteriormente, |
| Resumen | El sistema permite descargar los pdf generador de O.D.I en el perfil del trabajador |
| Postcondiciones | El O.D.I queda descargado en la máquina del usuario |
| Tipos | Principal – Documental – Financiero |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo de “Ver Trabajador” | 1. Muestra la vista del módulo |
| 1. Buscar trabajador por rut o nombre o cargo | 1. Lista los trabajadores en base a la petición |
| 1. Hace clic en el símbolo de descargar en la sección de “Registros de ODI” | 1. Descarga el archivo en la máquina del usuario |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.28: “Caso de uso N°46”*

| Caso de Uso N°46 | Descargando pdf de O.D.I filtrado con fecha |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permitir descargar el archivo pdf firmado de los O.D.I entregados entre fechas |
| Precondiciones | El usuario debe estar registrado anteriormente, deben existir archivos O.D.I |
| Resumen | El sistema permite descargar los pdf generados en O.D.I. |
| Postcondiciones | El O.D.I queda descargado en la máquina del usuario |
| Tipos | Principal – Documental – Financiero |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo de “Gestión de descargas” | 1. Muestra la vista del módulo |
| 1. Filtrar por tipo de archivo O.D.I y rango de fecha | 1. Lista los O.D.I en base a la petición |
| 1. Descarga el archivo en la zona de “Descargar” | 1. Envía el archivo al usuario y los descarga en su máquina |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.29: “Caso de uso N°47”*

| Caso de Uso N°47 | Descargando pdf de contrato filtrado con fecha |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permitir descargar el archivo pdf firmado de contrato entregados entre fechas |
| Precondiciones | El usuario debe estar registrado anteriormente, deben existir archivos contrato |
| Resumen | El sistema permite descargar los pdf generados en contrato |
| Postcondiciones | El contrato queda descargado en la máquina del usuario |
| Tipos | Principal – Documental – Financiero |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| 1. Accede al módulo de “Gestión de descargas” | 1. Muestra la vista del módulo |
| 1. Filtrar por tipo de archivo contrato y rango de fecha | 1. Lista los contrato en base a la petición |
| 1. Descarga el archivo en la zona de “Descargar” | 1. Envía el archivo al usuario y los descarga en su máquina |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

##### *Tabla 3.30: “Caso de uso N°48”*

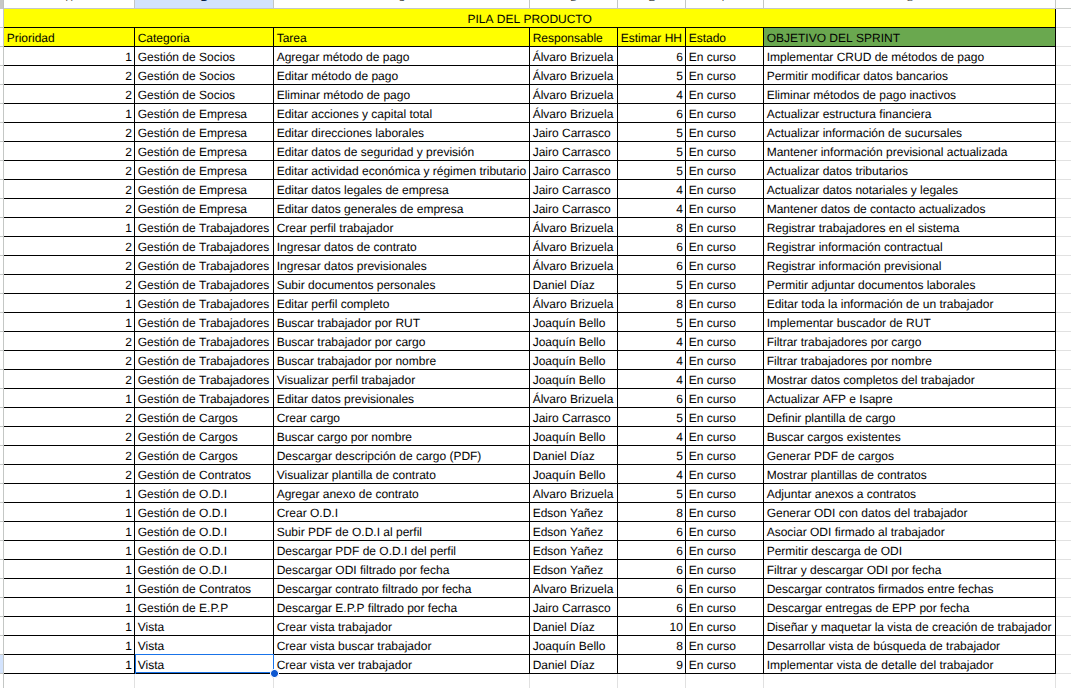
| Caso de Uso N°48 | Descargando pdf de E.P.P filtrado por fecha |
| --- | --- |
| Actores | Recursos Humanos, Administrador |
| Propósito | Permitir descargar el archivo pdf firmado de los E.P.P entregados entre fechas |
| Precondiciones | El usuario debe estar registrado anteriormente, deben existir archivos E.P.P en el perfil del trabajador |
| Resumen | El sistema permite descargar los pdf generador en asignando E.P.P al perfil del trabajador |
| Postcondiciones | El E.P.P queda descargado en la máquina del usuario |
| Tipos | Principal – Documental – Financiero |
| Curso normal de eventos | |
| Acción de los actores | Respuesta del sistema |
| Accede al módulo de “Gestión de descargas” | Muestra la vista del módulo |
| Filtrar por tipo de archivo EPP y rango de fecha | Lista los EPP en base a la petición |
| Descarga el archivo en la zona de “Descargar” | Envía el archivo al usuario y los descarga en su máquina |

Fuente: Tabla creada por los alumnos encargados del proyecto con fines académicos de la Universidad Andrés Bello como parte del curso Ingeniería de software (2025).

# 4. TABLA BACKLOG SIN TIEMPO REAL

A continuación, en la Tabla 4.1, se presenta el backlog estimado del producto, el cual reúne el conjunto de tareas planificadas para el desarrollo del sistema antes del inicio de su ejecución. Esta tabla representa la planificación inicial, estableciendo la prioridad, categoría funcional y responsables asignados para cada una de las tareas consideradas esenciales para alcanzar los objetivos del proyecto.

##### Tabla 4.1 – “Tabla backlog”



*Fuente: Elaborado por el estudiante de acuerdo con el proyecto*

# 

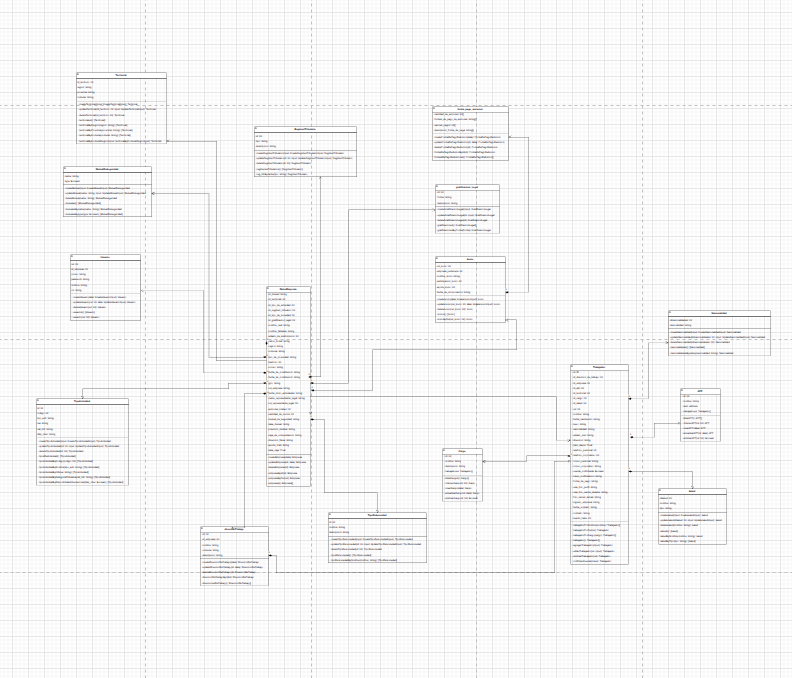
# 

# **5. DIAGRAMA DE CLASES**

A continuación, en la Figura 5.1, se presenta el Diagrama de Clases correspondiente a la arquitectura lógica del sistema. Esta figura representa la estructura fundamental de las entidades del dominio, incluyendo sus atributos, métodos y las relaciones existentes entre ellas. El propósito de este diagrama es modelar las clases que forman parte del sistema y establecer cómo interactúan entre sí mediante asociaciones, agregaciones o dependencias.

### 

### *Figura* 5.1 *“Diagrama de clases”*

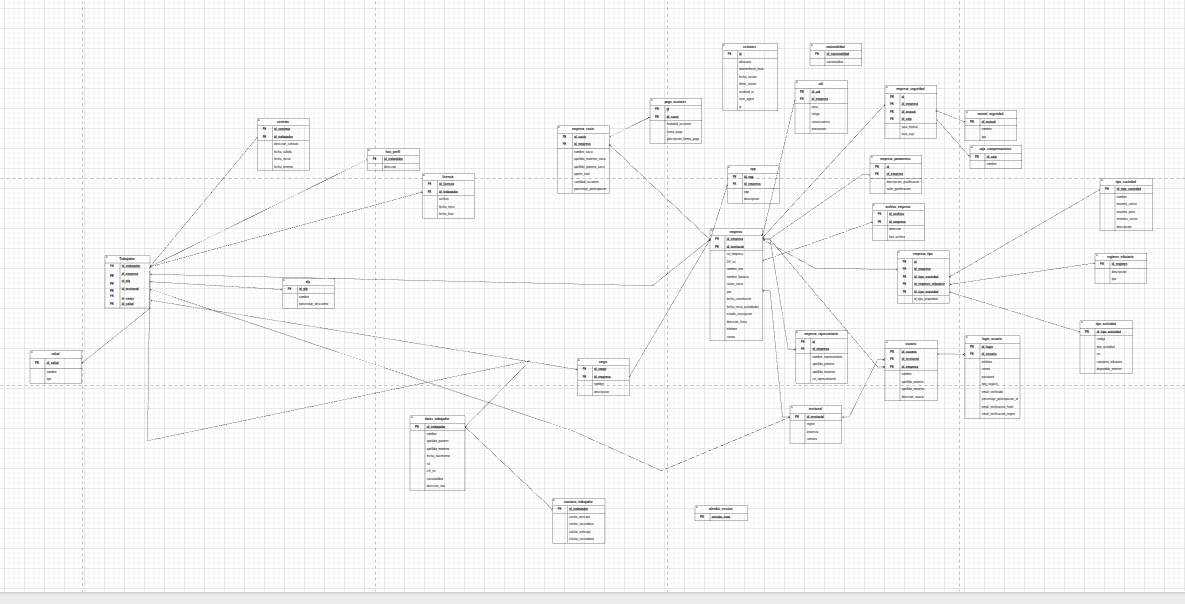


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# **6. DIAGRAMA DE DATOS**

A continuación, en la Figura 6.1, se presenta el Diagrama de Datos, el cual representa la estructura lógica de la base de datos diseñada para el sistema. En este diagrama se visualizan las principales tablas, sus respectivos atributos, tipos de datos y las relaciones existentes entre ellas mediante claves primarias y foráneas.

### *Figura* 6.1 *“Diagrama de datos”*



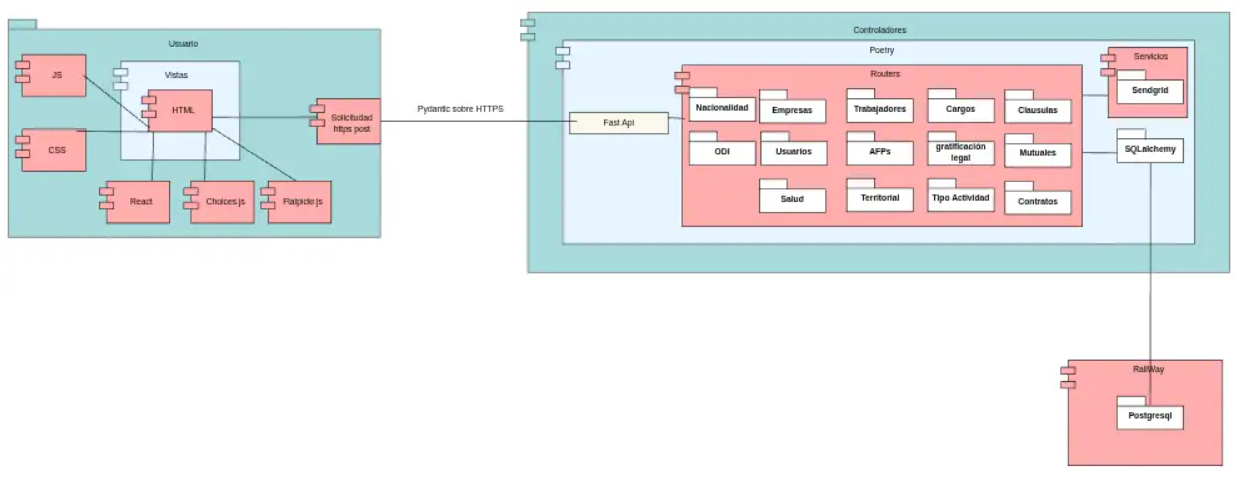
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# **7. DIAGRAMA DE COMPONENTES**

A continuación, en la Figura 7.1, se presenta el Diagrama de Componentes, el cual representa la estructura modular del sistema desde una perspectiva de alto nivel. En esta figura se identifican los principales bloques funcionales del sistema, junto con sus relaciones e interdependencias lógicas.

# 

### *Figura* 7.1 *“Diagrama de componentes”*

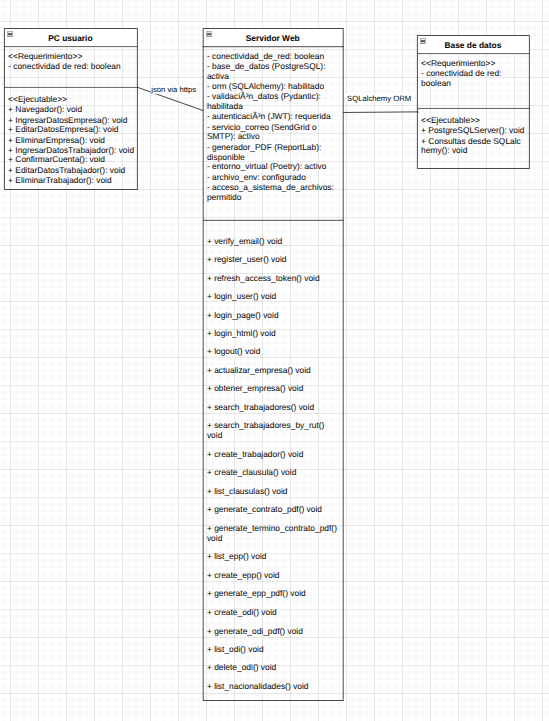


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# **8. DIAGRAMA DE DESPLIEGUE**

A continuación, en la Figura 8.1, se presenta el Diagrama de Despliegue, el cual describe la distribución física de los componentes del sistema sobre la infraestructura tecnológica disponible. Este diagrama especifica cómo se organiza el sistema en distintos nodos, detallando los elementos ejecutables y las conexiones entre ellos.

### *Figura* 8.1 *“Diagrama de despliegue”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# 9. DIAGRAMA DE SECUENCIA

Los siguientes diagramas de secuencia muestran la lógica del funcionamiento de los casos de uso en el sistema, a continuación desde la figura 9.1 hasta la 9.30 se puede ver el funcionamiento del sistema.

## **9.1 Caso de uso N°11**

En la figura 9.1 se muestra la interacción entre el Administrador y la vista V\_datos-empresa, junto con los componentes Pydantic, C\_edit\_socio, SQLAlchemy y las entidades empresa y socio durante el proceso de asociar un nuevo método de pago a un socio. El flujo inicia cuando el administrador accede a la sección de datos de la empresa y agrega la información bancaria o medio de pago correspondiente. Los datos son validados mediante Pydantic y enviados al router de edición, donde SQLAlchemy ejecuta una actualización (UPDATE) sobre las tablas de empresa y socios. Finalmente, el sistema devuelve una notificación de éxito, que se presenta en la interfaz como un mensaje de confirmación, indicando que el método de pago fue registrado correctamente.

### *Figura* 9.1 *“Diagrama de Secuencia - caso de uso 11”*

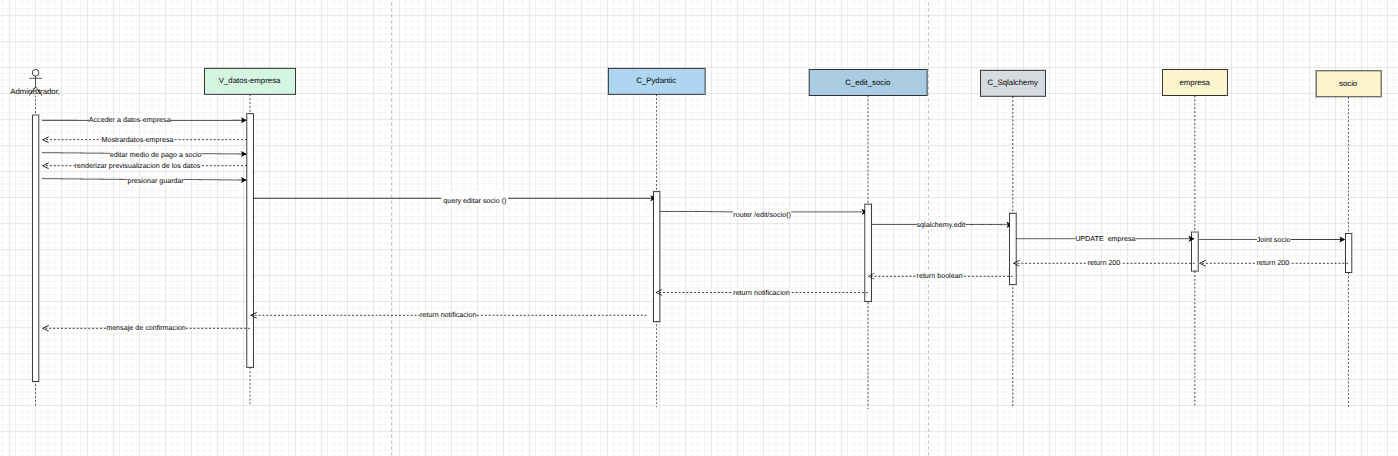


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.2 Caso de uso N°12**

En la figura 9.2 se presenta la secuencia de interacción entre el Administrador y la vista V\_datos-empresa, junto con los componentes Pydantic, C\_edit\_socio, SQLAlchemy y las entidades empresa y socio, durante el proceso de modificación de un método de pago asociado a un socio. El flujo inicia cuando el administrador accede a la sección de datos de la empresa, selecciona al socio correspondiente y edita la información del medio de pago. Los datos son validados por Pydantic y enviados al endpoint de actualización, donde SQLAlchemy ejecuta la instrucción UPDATE en la base de datos. Una vez completada la operación, el sistema retorna una notificación de éxito, reflejada en la interfaz mediante un mensaje de confirmación que indica que la modificación del método de pago se realizó correctamente.

### *Figura* 9.2 *“Diagrama de Secuencia - caso de uso 12”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## 

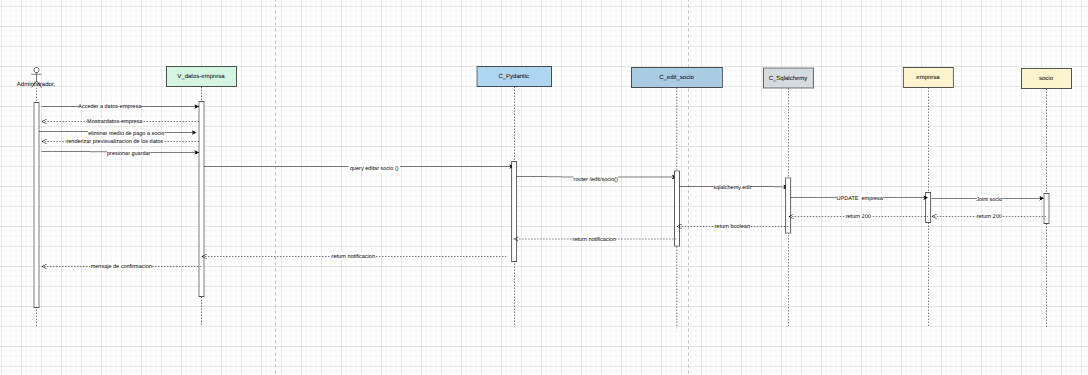
## 

## 

## **9.3 Caso de uso N°13**

En la figura 9.3 se muestra el flujo de interacción entre el Administrador y la vista V\_datos-empresa, junto con los componentes Pydantic, C\_edit\_socio, SQLAlchemy y las entidades empresa y socio, durante el proceso de eliminación de un método de pago asociado a un socio. El administrador accede a la sección de datos de empresa, selecciona al socio correspondiente y elimina el método de pago previamente registrado. La acción es validada por Pydantic y enviada al endpoint de actualización, donde SQLAlchemy ejecuta la modificación pertinente en la base de datos. Finalmente, el sistema retorna una notificación de confirmación, que se muestra en la interfaz indicando que el método de pago fue eliminado correctamente.

### *Figura* 9.3 *“Diagrama de Secuencia - caso de uso 13”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## 

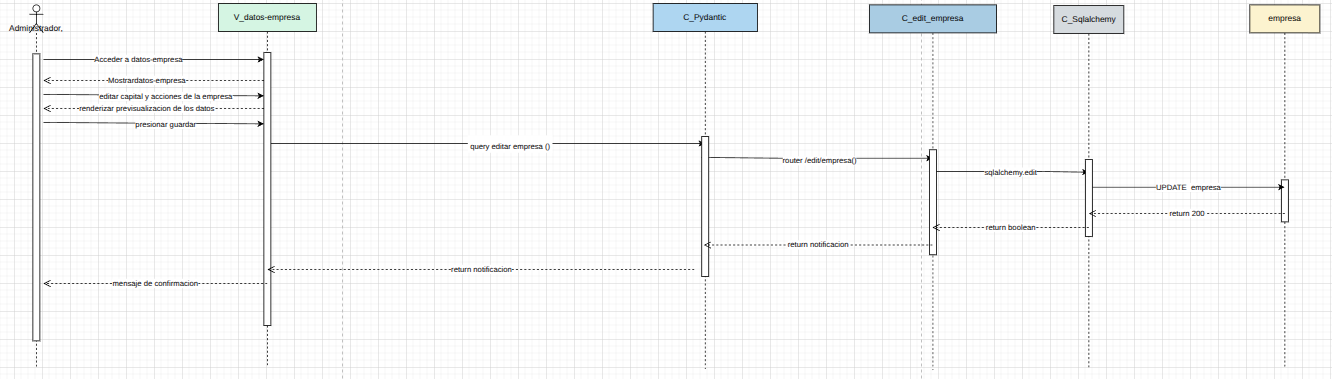
## 

## 

## **9.4 Caso de uso N°14**

En la figura 9.4 se muestra la interacción entre el Administrador y la vista V\_datos-empresa, junto con los componentes Pydantic, C\_edit\_empresa, SQLAlchemy y la entidad empresa, durante el proceso de modificación del capital y las acciones de la empresa. El flujo inicia cuando el administrador accede a la sección de datos empresariales, realiza los cambios en los campos correspondientes y visualiza una previsualización antes de guardar. Los datos son validados por Pydantic y enviados al endpoint de actualización, donde SQLAlchemy ejecuta la instrucción UPDATE sobre la tabla de empresa. Finalmente, el sistema devuelve una notificación de éxito, reflejada en la interfaz mediante un mensaje de confirmación que indica que la actualización del capital y las acciones se realizó correctamente.

### *Figura* 9.4 *“Diagrama de Secuencia - caso de uso 14”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## 

## 

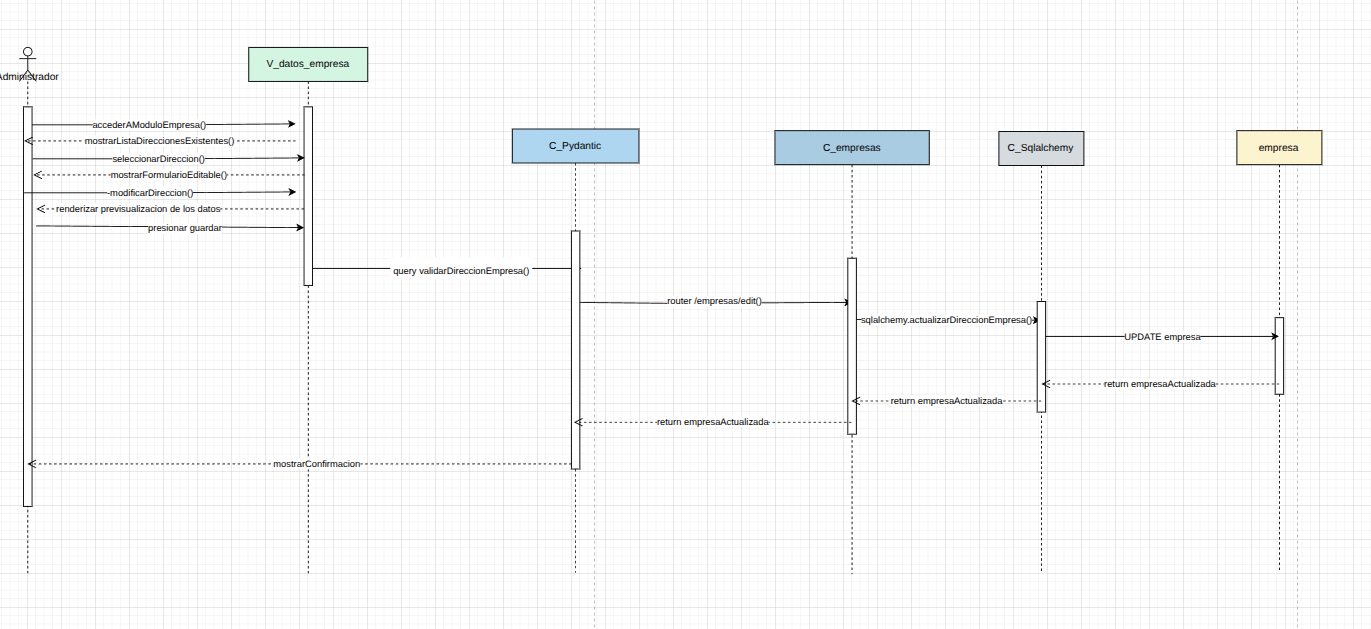
## 

## 

## **9.5 Caso de uso N°15**

En la figura 9.5 se muestra el proceso mediante el cual el Administrador accede al módulo V\_datos-empresa para modificar la dirección registrada de la empresa. El flujo comienza cuando el usuario ingresa al módulo, visualiza la lista de direcciones existentes y selecciona la que desea editar. Luego, el sistema muestra un formulario editable que permite realizar los cambios necesarios. Los datos modificados son validados por Pydantic y enviados al controlador C\_empresas, que utiliza SQLAlchemy para ejecutar la actualización (UPDATE) en la base de datos sobre la tabla correspondiente. Finalmente, se retorna una confirmación de éxito, la cual se refleja en la interfaz mediante un mensaje de confirmación que indica que la dirección fue actualizada correctamente.

### *Figura* 9.5 *“Diagrama de Secuencia - caso de uso 15”*

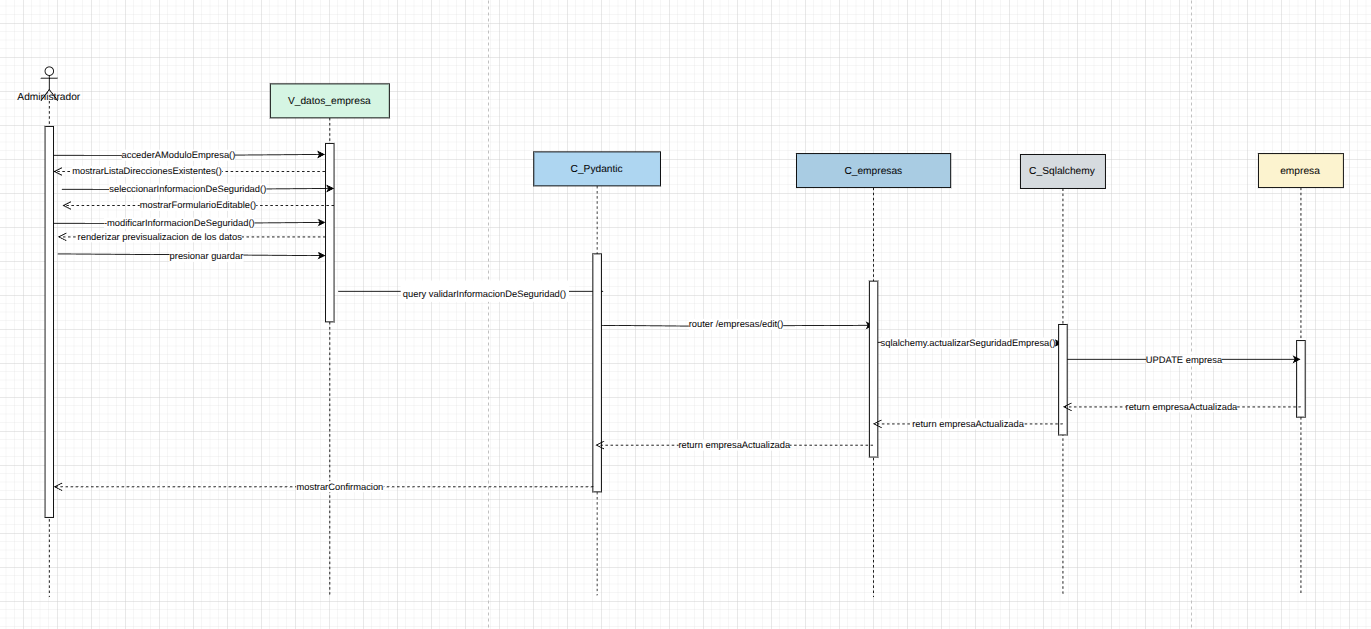


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.6 Caso de uso N°16**

En la figura 9.6 se representa el flujo de interacción entre el Administrador y la vista V\_datos\_empresa, junto con los componentes Pydantic, C\_empresas, SQLAlchemy y la entidad empresa, durante el proceso de modificación de la información de seguridad empresarial. El flujo inicia cuando el administrador accede al módulo de empresa, selecciona la información de seguridad que desea actualizar y realiza los cambios pertinentes. Los datos son validados por Pydantic y enviados al controlador C\_empresas, que a través de SQLAlchemy ejecuta la actualización (UPDATE) en la base de datos. Finalmente, el sistema retorna una confirmación de éxito, la cual se muestra en la interfaz como un mensaje informativo que indica que la información de seguridad fue actualizada correctamente.

### *Figura* 9.6 *“Diagrama de Secuencia - caso de uso 16”*



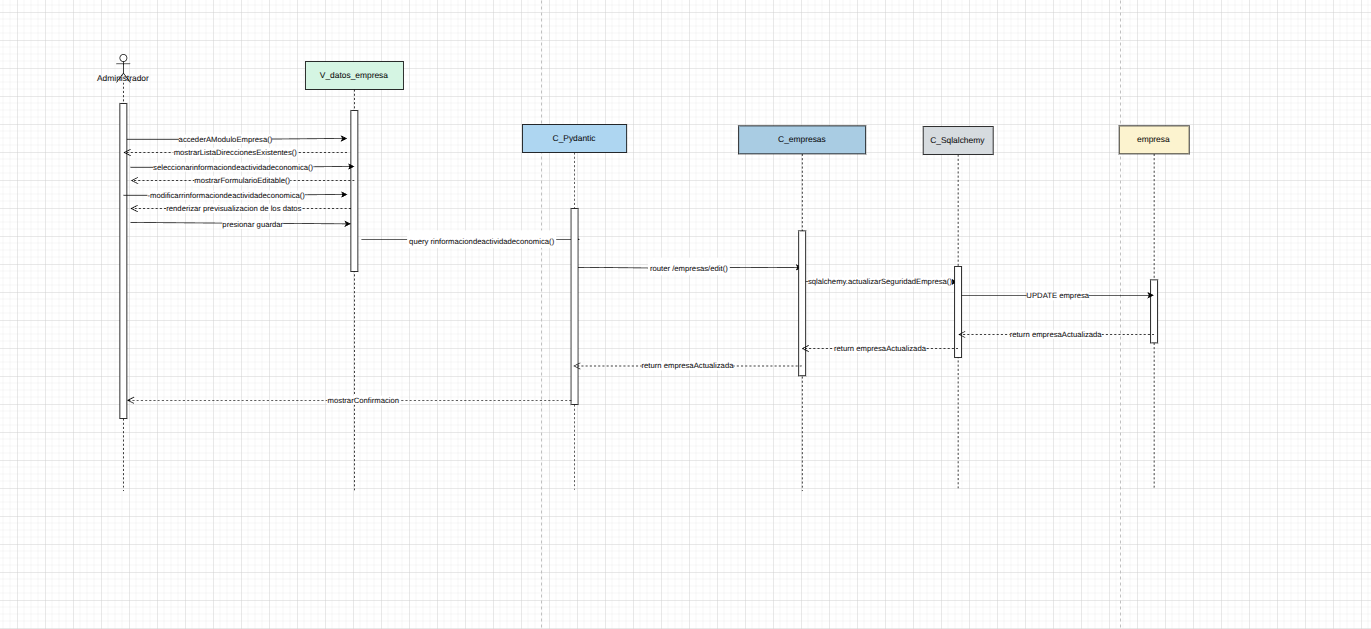
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.7 Caso de uso N°17**

Figura 9.7. Editando información de actividad económica de la empresa

En la figura 9.7 se detalla el flujo de interacción entre el Administrador y la vista V\_datos\_empresa, junto con los componentes Pydantic, C\_empresas, SQLAlchemy y la entidad empresa, durante el proceso de actualización de la información de actividad económica. El proceso comienza cuando el administrador accede al módulo de empresa, selecciona la información económica que desea modificar y realiza los ajustes necesarios en el formulario editable. Los datos son validados por Pydantic y enviados al controlador C\_empresas, que a través de SQLAlchemy ejecuta la instrucción UPDATE en la base de datos. Finalmente, el sistema devuelve una confirmación de éxito, reflejada en la interfaz mediante un mensaje que indica que la información económica de la empresa fue actualizada correctamente.

### *Figura* 9.7 *“Diagrama de Secuencia - caso de uso 17”*



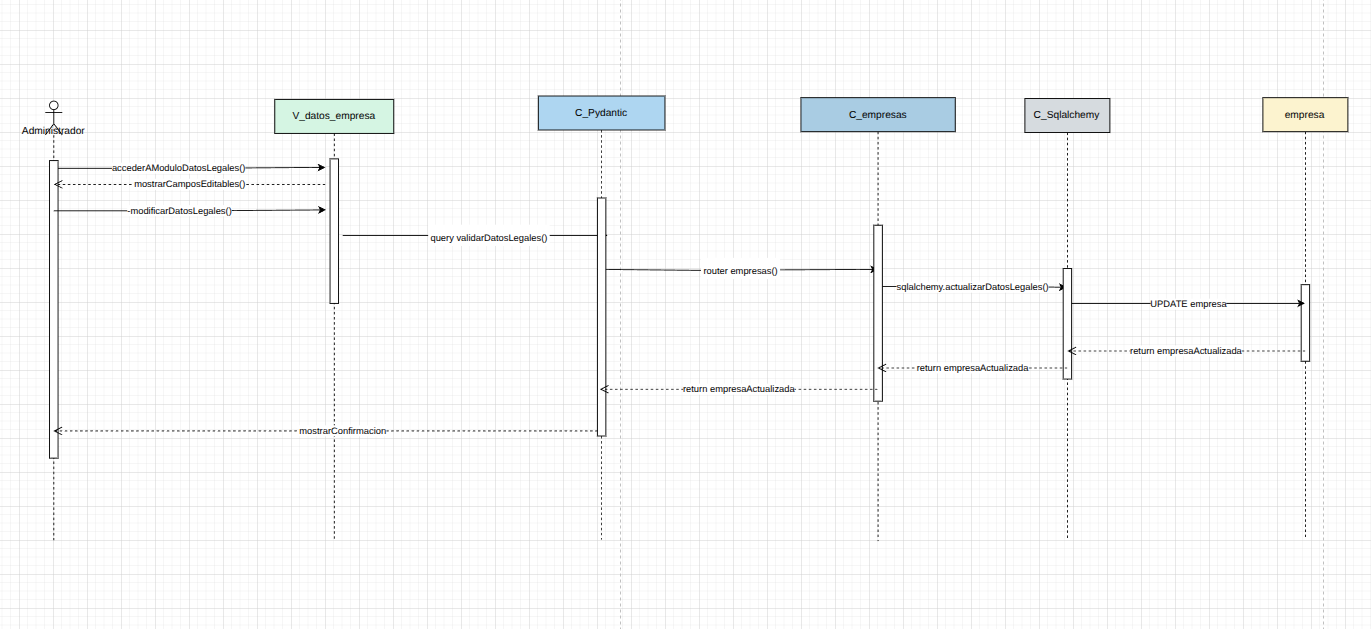
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## **9.8 Caso de uso N°18**

En la figura 9.8 se observa el flujo de interacción entre el Administrador y la vista V\_datos\_empresa, junto con los componentes Pydantic, C\_empresas, SQLAlchemy y la entidad empresa, durante el proceso de actualización de los datos legales de la empresa. El flujo comienza cuando el administrador accede al módulo de datos legales, donde puede visualizar y modificar los campos editables correspondientes a la razón social, tipo de sociedad u otros aspectos jurídicos. Los datos ingresados son validados por Pydantic y enviados al controlador C\_empresas, que mediante SQLAlchemy ejecuta la instrucción UPDATE en la base de datos. Una vez finalizado el proceso, el sistema devuelve una confirmación de éxito, mostrando en la interfaz un mensaje que indica que los datos legales de la empresa se actualizaron correctamente.

### *Figura* 9.8 *“Diagrama de Secuencia - caso de uso 18”*

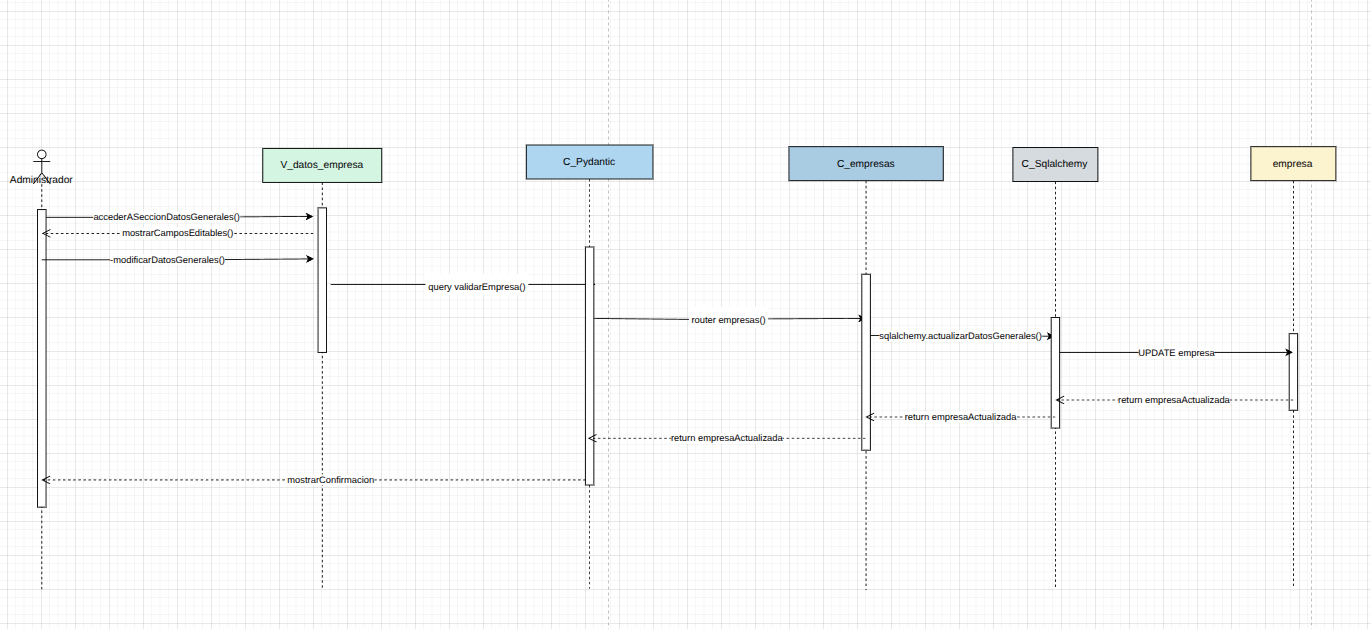


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.9 Caso de uso N°19**

En la figura 9.9 se ilustra el flujo de interacción entre el Administrador y la vista V\_datos\_empresa, junto con los componentes Pydantic, C\_empresas, SQLAlchemy y la entidad empresa, durante el proceso de modificación de los datos generales. El flujo inicia cuando el administrador accede a la sección de datos generales de la empresa y edita la información básica, como nombre, razón social o fecha de constitución. Los cambios son validados por Pydantic y enviados al controlador C\_empresas, el cual utiliza SQLAlchemy para ejecutar la instrucción UPDATE en la base de datos. Finalmente, el sistema retorna una confirmación de éxito, mostrando en la interfaz un mensaje que informa que los datos generales de la empresa fueron actualizados correctamente.

### *Figura* 9.9 *“Diagrama de Secuencia - caso de uso 19”*

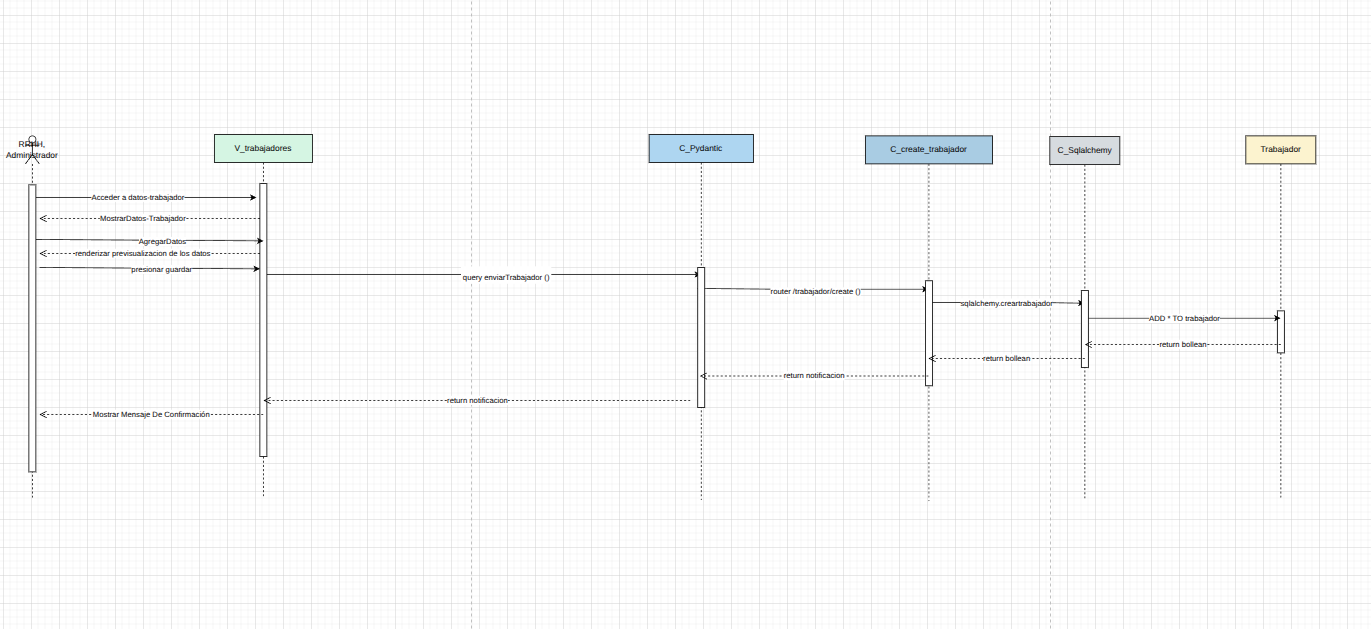


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.10 Caso de uso N°20**

En la figura 9.10 se presenta el proceso mediante el cual el Administrador o el área de Recursos Humanos (RRHH) accede al módulo de trabajadores (V\_trabajadores) para registrar un nuevo empleado en el sistema. El flujo inicia cuando el usuario ingresa los datos del trabajador y revisa la previsualización de la información antes de guardar. Los datos son validados por Pydantic y enviados al controlador C\_create\_trabajador, que a su vez utiliza SQLAlchemy para ejecutar la instrucción ADD sobre la tabla de trabajador. Una vez confirmada la creación, el sistema retorna una notificación de éxito, que se muestra en la interfaz mediante un mensaje de confirmación indicando que el nuevo trabajador fue registrado correctamente.

### *Figura* 9.10 *“Diagrama de Secuencia - caso de uso 20”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

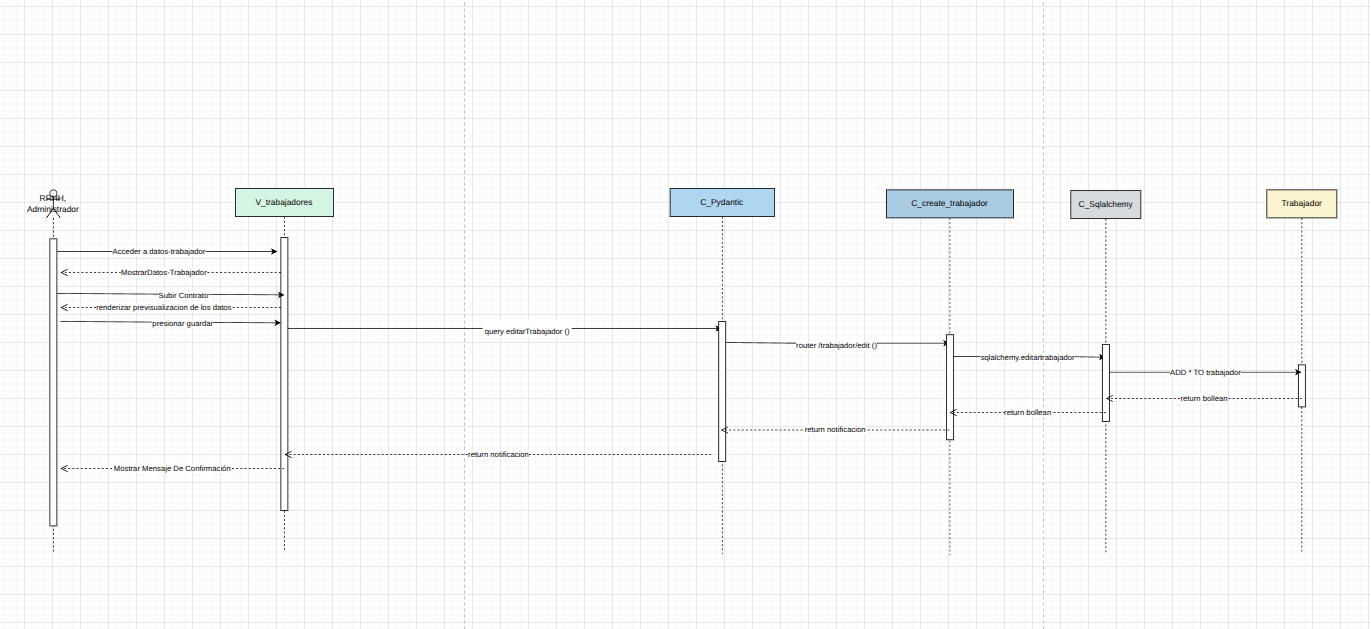
## 

## 

## **9.11 Caso de uso N°21**

En la figura 9.11 se muestra el flujo de interacción entre el Administrador o el área de Recursos Humanos (RRHH) y la vista V\_trabajadores, junto con los componentes Pydantic, C\_create\_trabajador, SQLAlchemy y la entidad Trabajador, durante el proceso de subida de un contrato laboral. El flujo inicia cuando el usuario accede a los datos del trabajador y selecciona la opción para subir el contrato. Los datos son validados mediante Pydantic y enviados al endpoint de edición, donde SQLAlchemy realiza la actualización correspondiente en la tabla del trabajador. Finalmente, el sistema retorna una notificación de confirmación, reflejada en la interfaz con un mensaje que indica que el contrato fue cargado exitosamente.

### *Figura* 9.11 *“Diagrama de Secuencia - caso de uso 21”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

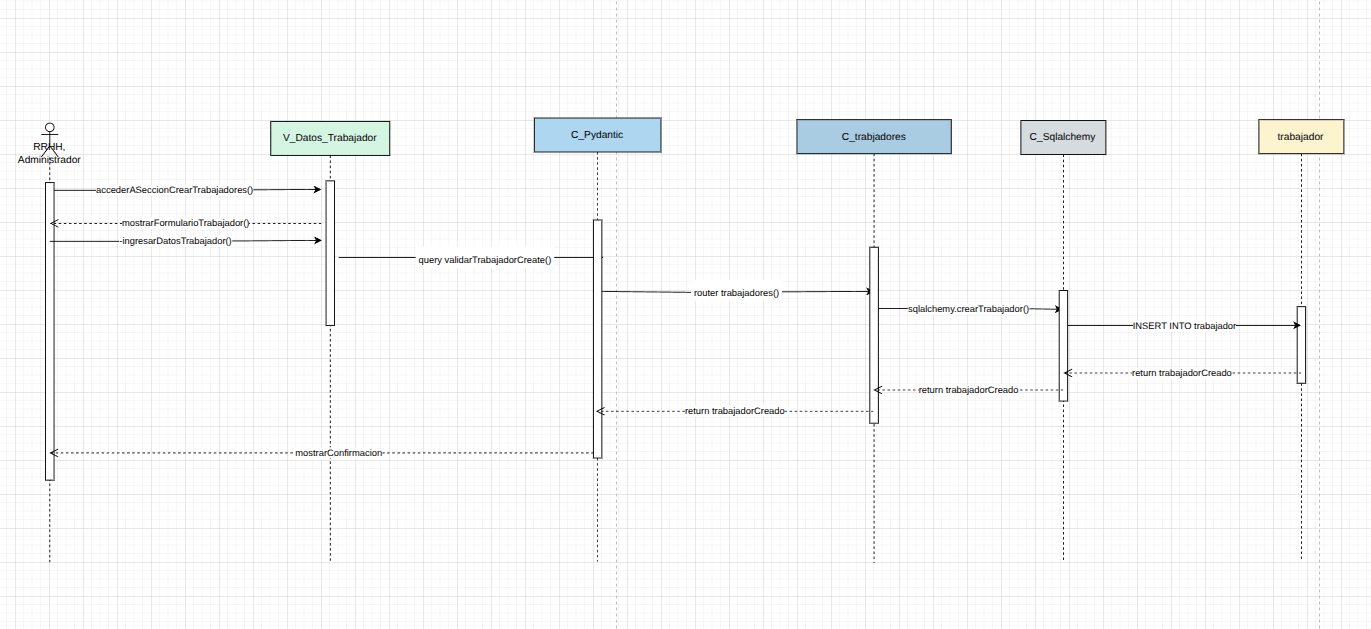
## 

## **9.12 Caso de uso N°22**

En la figura 9.12 se detalla el flujo de interacción correspondiente al proceso de creación de un nuevo trabajador desde el módulo de gestión de personal. El Administrador o el área de Recursos Humanos (RRHH) accede a la vista V\_Datos\_Trabajador, donde se despliega un formulario para ingresar los datos del nuevo empleado. Una vez completada la información, esta es validada mediante Pydantic, que verifica su estructura y consistencia antes de enviarla al controlador C\_trabajadores.

El controlador ejecuta la función crearTrabajador() utilizando SQLAlchemy, que realiza la instrucción INSERT INTO trabajador para registrar los datos en la base de datos. Tras la creación exitosa, el sistema retorna la confirmación de inserción y muestra al usuario un mensaje de confirmación, indicando que el trabajador fue añadido correctamente al sistema.

### *Figura* 9.12 *“Diagrama de Secuencia - caso de uso 22”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

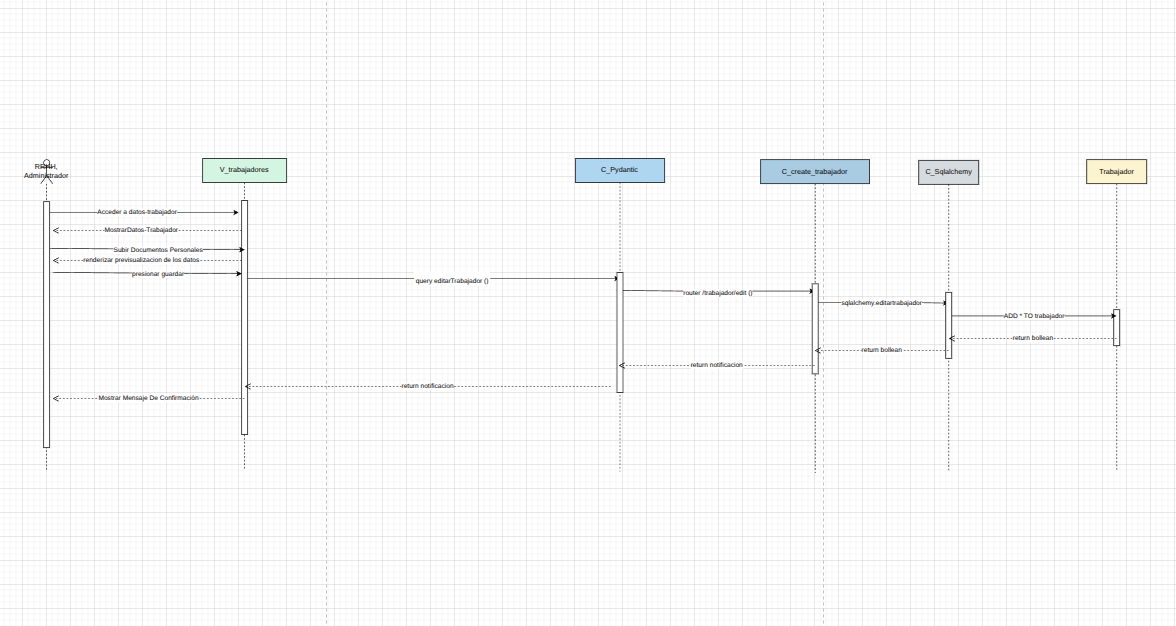
## **9.13 Caso de uso N°23**

En la figura 9.13 se representa el proceso mediante el cual el Administrador o el área de Recursos Humanos (RRHH) gestiona la subida de documentos personales de un trabajador dentro del módulo de empleados.

El flujo comienza cuando el usuario accede a la vista V\_trabajadores, donde puede seleccionar la opción para cargar documentos como identificaciones, certificados o comprobantes. Estos datos son validados por Pydantic antes de ser enviados al controlador C\_create\_trabajador, que utiliza SQLAlchemy para ejecutar la operación UPDATE sobre la tabla trabajador, integrando los archivos y metadatos correspondientes.

Una vez completada la actualización, el sistema devuelve una notificación de éxito, y la vista muestra un mensaje de confirmación, indicando que los documentos personales fueron correctamente almacenados y asociados al trabajador.

### *Figura* 9.13 *“Diagrama de Secuencia - caso de uso 23”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.14 Caso de uso N°24**

En la figura 9.14 se presenta el flujo correspondiente a la edición de los datos de un trabajador dentro del sistema. El proceso inicia cuando el Administrador o el área de Recursos Humanos (RRHH) accede a la vista V\_trabajadores, donde se muestran los datos actuales del empleado. Desde esta interfaz, el usuario puede modificar información personal, laboral o contractual del trabajador.

Una vez realizada la edición, los datos son validados mediante Pydantic para asegurar su consistencia y formato correcto. Posteriormente, la solicitud se envía al controlador C\_create\_trabajador, que a su vez utiliza SQLAlchemy para ejecutar una actualización (UPDATE) sobre la tabla trabajador.

Finalmente, el sistema confirma la operación devolviendo una notificación de éxito, la cual se refleja en la interfaz con un mensaje de confirmación, indicando que los cambios en los datos del trabajador fueron guardados satisfactoriamente.

### *Figura* 9.14 *“Diagrama de Secuencia - caso de uso 24”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## 

## **9.15 Caso de uso N°25**

En la figura 9.15 se describe el proceso de búsqueda de un trabajador registrado dentro del sistema. El flujo comienza cuando el Administrador o el área de Recursos Humanos (RRHH) accede a la vista V\_lista\_trabajadores, desde donde se puede visualizar el listado completo de empleados.

Al ingresar un RUT o identificador específico en el campo de búsqueda, la información es validada por Pydantic, que asegura la estructura correcta de la consulta. Luego, la solicitud es enviada al controlador C\_trabajadores\_search, el cual ejecuta a través de SQLAlchemy la instrucción SELECT \* FROM datos\_trabajador, filtrando los registros que coincidan con el criterio ingresado.

El resultado de la búsqueda se devuelve al controlador y posteriormente a la vista, donde se muestra el trabajador coincidente en la interfaz. Este proceso permite una búsqueda precisa y rápida dentro del módulo de gestión de personal, optimizando la localización de información específica en la base de datos.

### *Figura* 9.15 *“Diagrama de Secuencia - caso de uso 25”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.16 Caso de uso N°26**

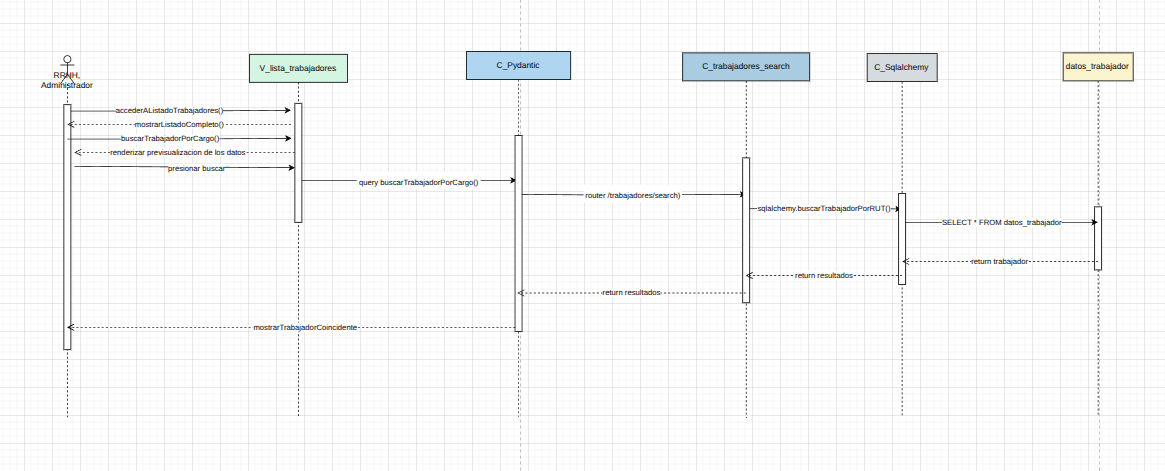
En la figura 9.16 se representa el flujo correspondiente a la búsqueda de trabajadores según su cargo dentro del sistema.

El proceso comienza cuando el Administrador o el área de Recursos Humanos (RRHH) accede a la vista V\_lista\_trabajadores, donde puede observar el listado general de empleados registrados. Posteriormente, el usuario introduce el nombre del cargo en el campo de búsqueda, lo que genera una consulta que es validada por Pydantic para asegurar el formato correcto.

A continuación, el controlador C\_trabajadores\_search recibe la petición y, utilizando SQLAlchemy, ejecuta una instrucción SELECT \* FROM datos\_trabajador filtrada por el campo cargo. Los resultados obtenidos son enviados nuevamente al controlador y retornados a la vista, donde se muestran en pantalla los trabajadores que coinciden con el cargo buscado.

De esta manera, el sistema facilita la localización rápida y eficiente del personal según su puesto laboral, permitiendo al usuario acceder directamente a la información relevante del área o función requerida.

### *Figura* 9.16 *“Diagrama de Secuencia - caso de uso 26”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

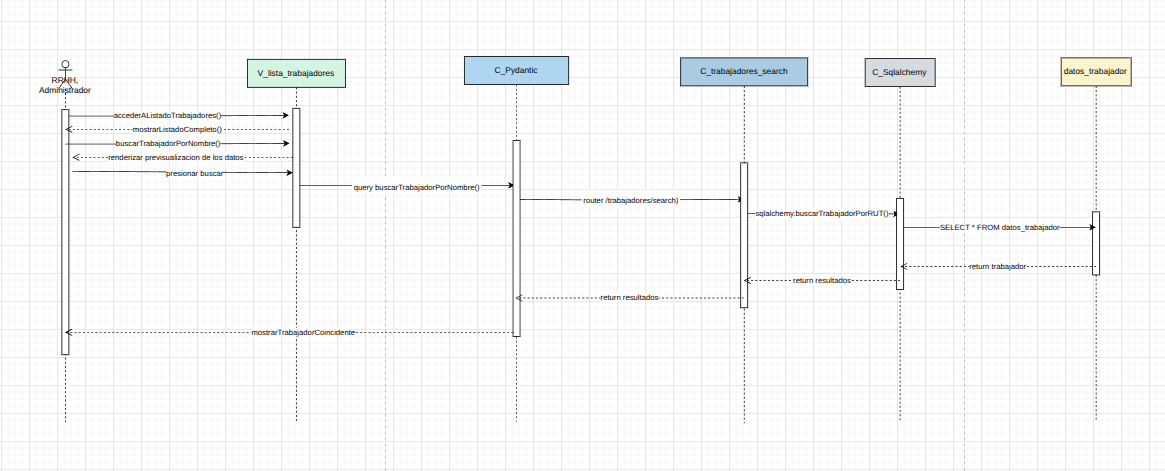
## **9.17 Caso de uso N°27**

La figura 9.17 muestra el proceso de búsqueda de trabajadores por nombre dentro del sistema. Este flujo inicia cuando el Administrador o el área de Recursos Humanos (RRHH) accede a la vista V\_lista\_trabajadores, donde se despliega el listado completo de los empleados registrados.

El usuario ingresa el nombre o parte del nombre del trabajador a buscar, y la solicitud es enviada al validador Pydantic, encargado de verificar la estructura y tipo de datos del campo ingresado. Posteriormente, el controlador C\_trabajadores\_search gestiona la búsqueda en la base de datos mediante SQLAlchemy, ejecutando una instrucción SELECT \* FROM datos\_trabajador con un filtro aplicado sobre el campo nombre.

Finalmente, el resultado es devuelto a la vista, donde se muestra el trabajador o los trabajadores coincidentes, permitiendo al usuario acceder de forma ágil a la información del personal requerido sin necesidad de recorrer manualmente toda la lista.

### *Figura* 9.17 *“Diagrama de Secuencia - caso de uso 27”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.18 Caso de uso N°28**

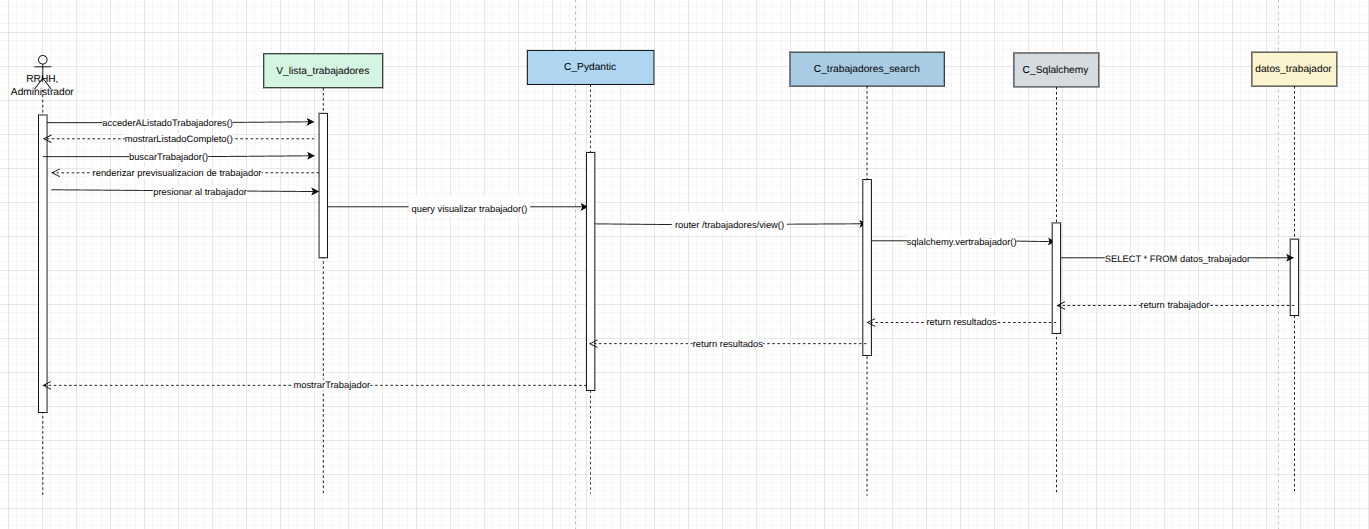
La figura 9.18 representa el proceso mediante el cual el Administrador o el área de Recursos Humanos (RRHH) accede a la información completa de un trabajador dentro del sistema.

El flujo comienza cuando el usuario entra al listado general de empleados desde la vista V\_lista\_trabajadores, donde puede visualizar una previsualización básica de los datos. Al seleccionar un trabajador específico, se ejecuta la función visualizarTrabajador(), la cual envía la solicitud al esquema de validación Pydantic.

Luego, el controlador C\_trabajadores\_search procesa la petición y utiliza SQLAlchemy para ejecutar una consulta SELECT \* FROM datos\_trabajador, recuperando toda la información relacionada con el empleado seleccionado.

Finalmente, los resultados son devueltos a la vista, donde se muestran los datos completos del trabajador, tales como identificación, cargo, contrato, historial y otros atributos relevantes. Este proceso garantiza una consulta rápida y organizada de la información del personal, centralizando los datos en un único flujo coherente y accesible.

### *Figura* 9.18 *“Diagrama de Secuencia - caso de uso 28”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.19 Caso de uso N°29**

La figura 9.19 representa el proceso mediante el cual el Administrador o el área de Recursos Humanos (RRHH) actualiza la información de un trabajador dentro del sistema.

El flujo inicia cuando el usuario accede a la sección de gestión de trabajadores (V\_trabajadores) y selecciona al empleado cuyos datos desea modificar. En la interfaz se muestran los datos actuales, permitiendo su edición. Una vez realizados los cambios, el usuario presiona el botón “Guardar”, lo que desencadena la consulta editarTrabajador().

Esta solicitud es enviada al validador Pydantic, que verifica la estructura y tipos de datos antes de dirigir la información al controlador C\_create\_trabajador. Este controlador utiliza SQLAlchemy para ejecutar la instrucción de actualización correspondiente sobre la entidad Trabajador.

Finalmente, el sistema retorna una notificación indicando el resultado de la operación y muestra un mensaje de confirmación en la interfaz, informando al usuario que los datos del trabajador fueron actualizados correctamente en la base de datos.

### *Figura* 9.19 *“Diagrama de Secuencia - caso de uso 29”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## **9.20 Caso de uso N°30**

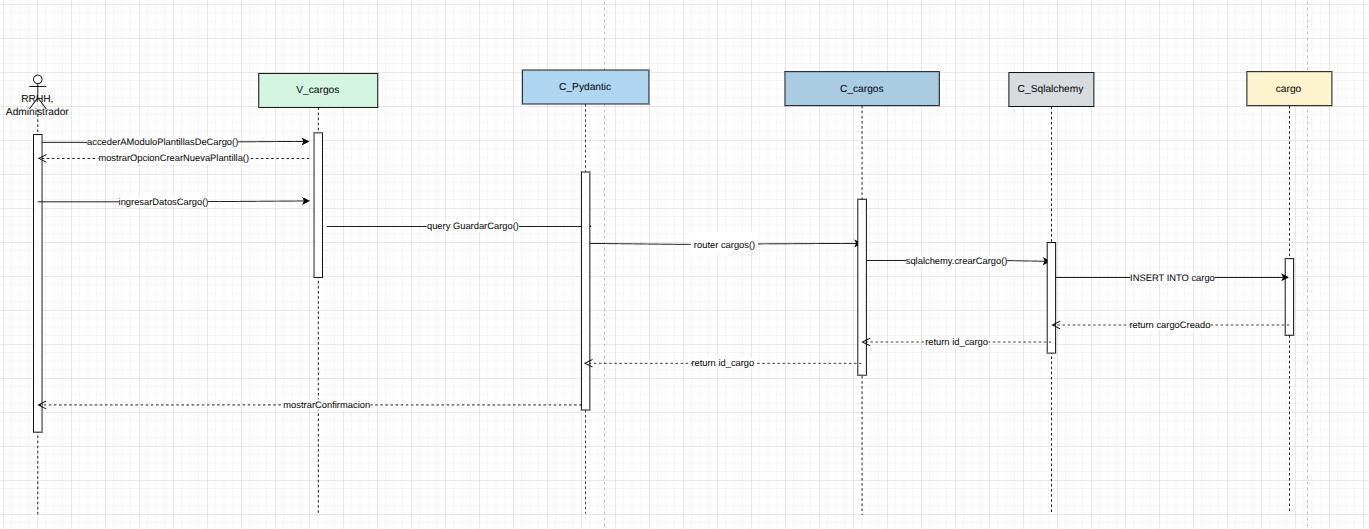
La figura 9.20 ilustra el flujo de creación de un nuevo cargo dentro del sistema, acción ejecutada por el Administrador o el área de Recursos Humanos (RRHH).

El proceso comienza cuando el usuario accede al módulo V\_cargos, donde el sistema muestra la opción para crear una nueva plantilla de cargo. El usuario ingresa los datos requeridos —como nombre del cargo, descripción, funciones y nivel jerárquico—, tras lo cual se ejecuta la consulta GuardarCargo().

El esquema Pydantic valida la estructura y tipos de datos, enviando la información al controlador C\_cargos, que a su vez llama al método crearCargo() de SQLAlchemy. Este último ejecuta la instrucción INSERT INTO cargo, registrando el nuevo cargo en la base de datos.

Una vez finalizada la operación, se retorna el id del nuevo cargo al controlador y posteriormente a la vista, donde se muestra un mensaje de confirmación notificando que el cargo fue creado exitosamente y ya está disponible para su asignación a trabajadores o plantillas futuras.

### *Figura* 9.20 *“Diagrama de Secuencia - caso de uso 30”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.21 Caso de uso N°31**

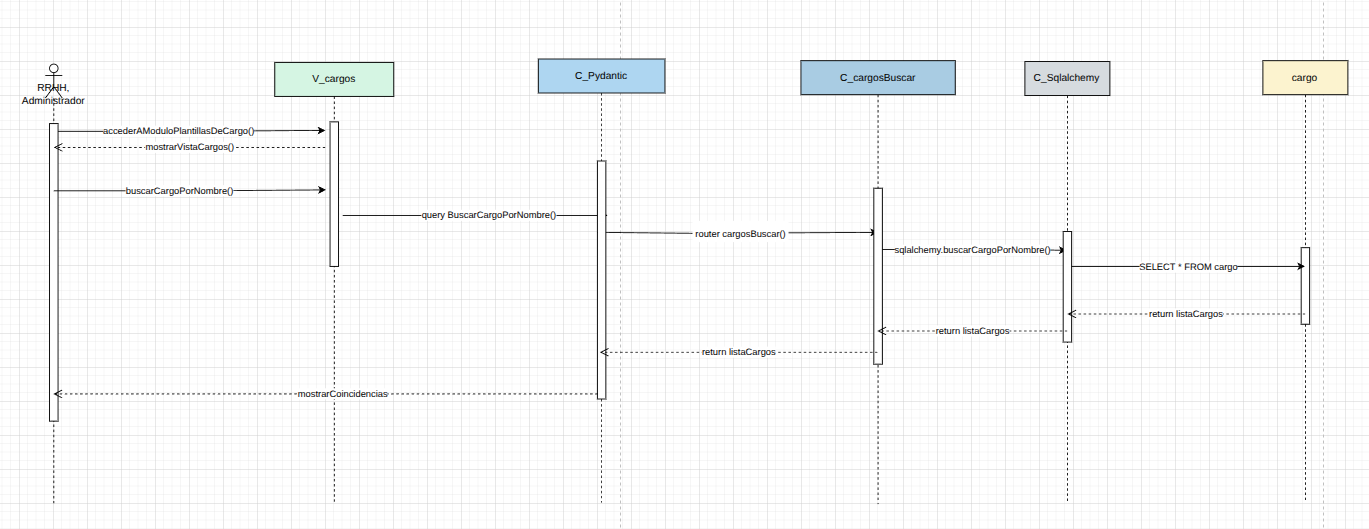
La figura 9.21 describe el proceso de búsqueda de un cargo dentro del sistema, acción que realiza el Administrador o el área de Recursos Humanos (RRHH).

El flujo comienza cuando el usuario accede al módulo de plantillas de cargos mediante la vista V\_cargos, donde el sistema muestra la interfaz con la opción de búsqueda. Al ingresar el nombre del cargo a localizar, se ejecuta la función buscarCargoPorNombre(), que envía una consulta de tipo BuscarCargoPorNombre().

El validador Pydantic recibe los datos de entrada y los transmite al controlador C\_cargosBuscar, el cual se encarga de ejecutar la función sqlalchemy.buscarCargoPorNombre(). Esta operación realiza una consulta SELECT \* FROM cargo en la base de datos, filtrando por el nombre especificado.

Una vez recuperados los resultados, la lista de coincidencias se devuelve al usuario a través del flujo inverso, siendo procesada por la vista para mostrar los cargos coincidentes. De esta forma, el sistema permite una búsqueda rápida y eficiente dentro del catálogo de cargos disponibles.

### *Figura* 9.21 *“Diagrama de Secuencia - caso de uso 31”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.22 Caso de uso N°32**

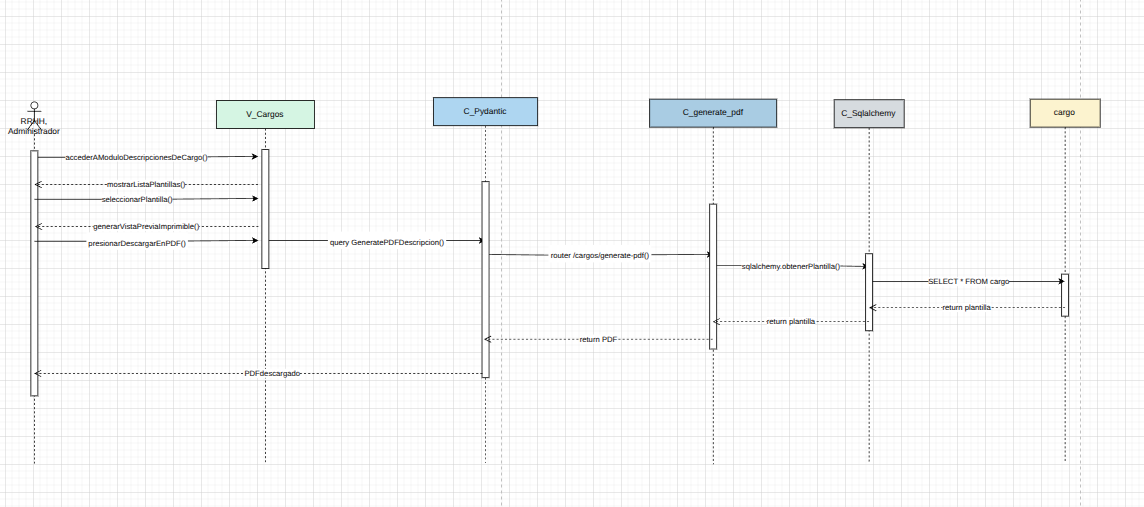
La figura 9.22 muestra el flujo del proceso mediante el cual el Administrador o el área de Recursos Humanos (RRHH) genera y descarga en formato PDF la descripción de un cargo previamente registrado en el sistema.

El proceso comienza cuando el usuario accede al módulo de descripciones de cargo desde la vista V\_Cargos, donde el sistema presenta la lista de plantillas disponibles. Luego, el usuario selecciona una plantilla específica, visualiza una vista previa imprimible, y finalmente presiona la opción “Descargar en PDF”.

La acción desencadena la consulta GeneratePDFDescripcion(), la cual es validada por el esquema Pydantic antes de ser enviada al controlador C\_generate\_pdf. Este controlador ejecuta la función sqlalchemy.obtenerPlantilla(), que realiza una consulta SELECT \* FROM cargo para recuperar la información del cargo correspondiente.

Una vez obtenidos los datos, el sistema genera el archivo PDF con la descripción completa del cargo (nombre, funciones, responsabilidades y requisitos), retornando el documento a la vista para su descarga directa por el usuario, confirmando el evento con el mensaje PDF descargado.

### *Figura* 9.22 *“Diagrama de Secuencia - caso de uso 32”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.23 Caso de uso N°34**

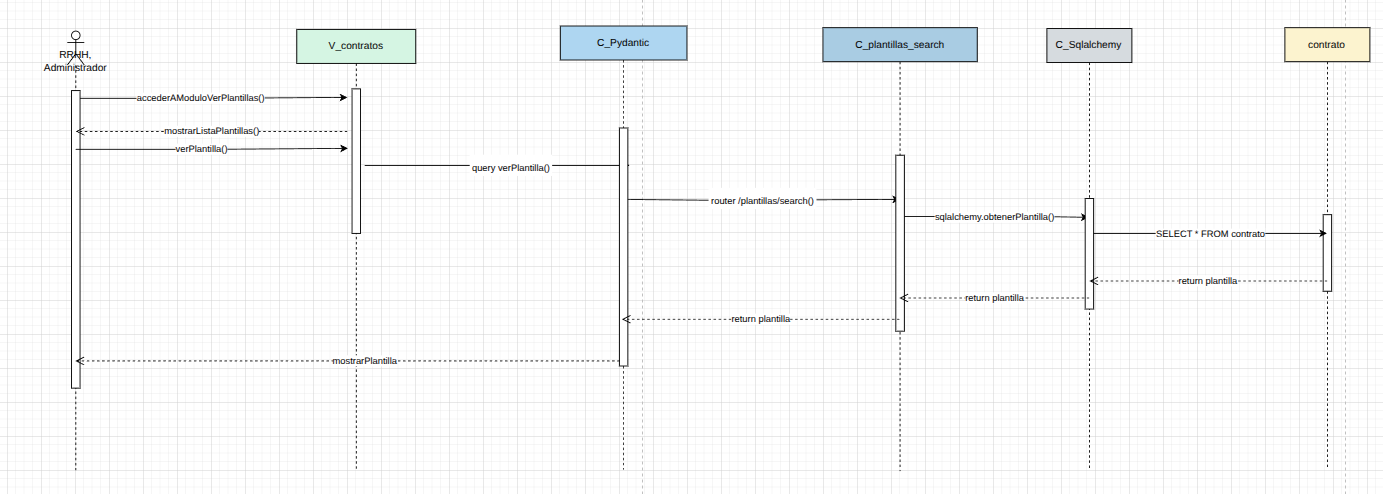
La figura 9.23 muestra el flujo de interacción para visualizar una plantilla de contrato dentro del sistema, funcionalidad utilizada por el Administrador o el área de Recursos Humanos (RRHH).

El proceso se inicia cuando el usuario accede al módulo correspondiente a las plantillas de contrato mediante la vista V\_contratos, donde el sistema muestra la lista de plantillas disponibles. Posteriormente, el usuario selecciona una plantilla específica mediante la acción verPlantilla(), la cual envía la solicitud query verPlantilla() al esquema Pydantic para su validación.

Una vez verificada la estructura de los datos, se activa el controlador C\_plantillas\_search, que ejecuta el método sqlalchemy.obtenerPlantilla() para consultar los registros en la base de datos mediante la instrucción SELECT \* FROM contrato.

El sistema retorna la plantilla seleccionada con todos sus detalles (tipo de contrato, cláusulas, condiciones y formato) y la muestra en la interfaz, completando el flujo con la acción mostrarPlantilla, que permite al usuario visualizar o editar la información antes de generar un nuevo contrato o guardarlo como referencia.

### *Figura* 9.23 *“Diagrama de Secuencia - caso de uso 34”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.24 Caso de uso N°38**

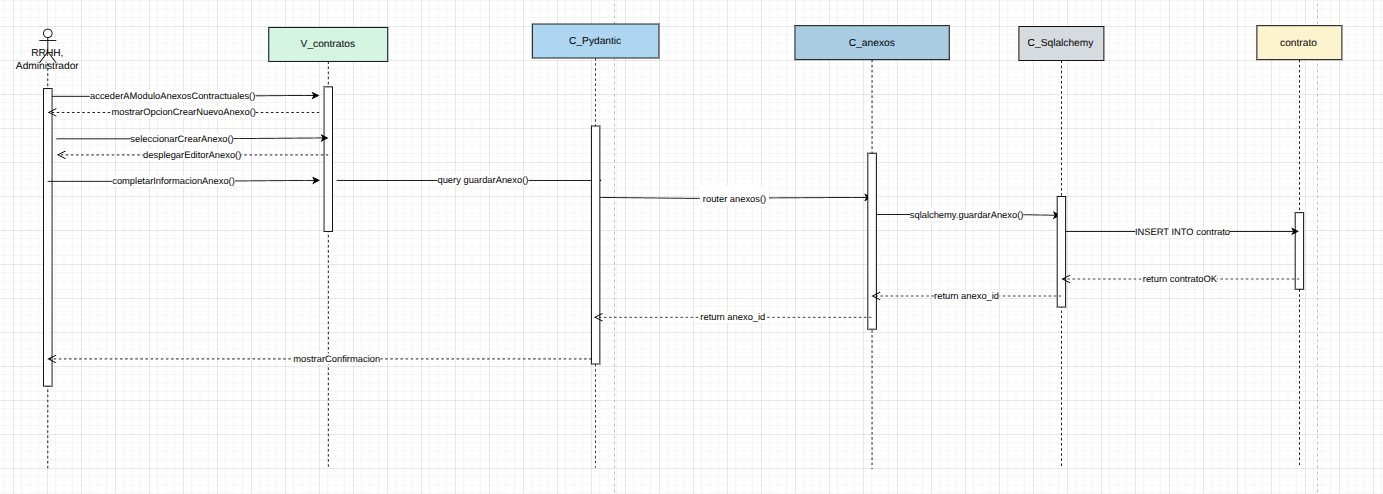
La figura 9.24 muestra el flujo correspondiente al proceso de creación de un nuevo anexo contractual, funcionalidad ejecutada por el Administrador o el área de Recursos Humanos (RRHH).

El proceso se inicia cuando el usuario accede al módulo de anexos contractuales mediante la vista V\_contratos, donde el sistema despliega la opción “Crear nuevo anexo”. Al seleccionar esta opción, se abre el editor de anexos, permitiendo al usuario completar la información necesaria (tipo de anexo, fecha de vigencia, cláusulas y observaciones).

Una vez completados los datos, se ejecuta la acción completarInformacionAnexo(), que envía la solicitud query guardarAnexo() al esquema Pydantic para su validación. Tras verificar la estructura, la solicitud es enviada al controlador C\_anexos, el cual llama al método sqlalchemy.guardarAnexo() para registrar la información en la base de datos mediante la instrucción INSERT INTO contrato.

El sistema confirma la inserción devolviendo un identificador único (anexo\_id) y una respuesta de éxito (contratoOK). Finalmente, la vista muestra un mensaje de confirmación, indicando que el nuevo anexo contractual fue creado correctamente y está disponible para futuras consultas o modificaciones.

### *Figura* 9.24 *“Diagrama de Secuencia - caso de uso 38”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.25 Caso de uso N°43**

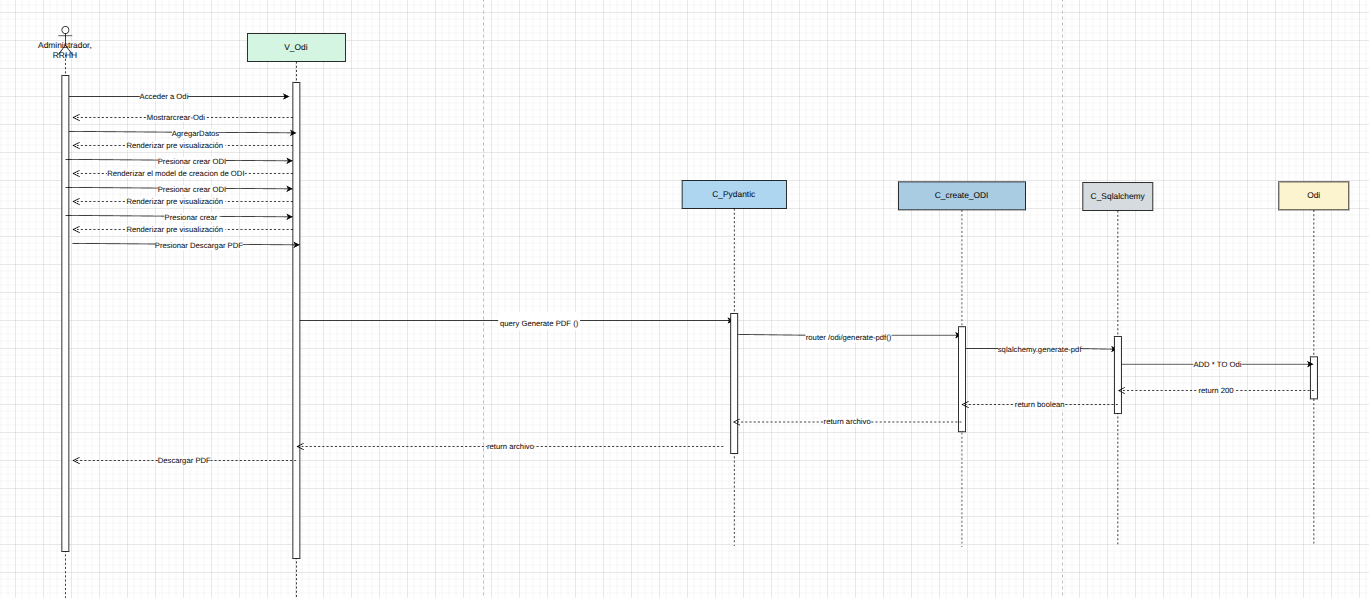
La figura 9.25 ilustra el flujo de creación y descarga del documento Orden de Ingreso (ODI) por parte del Administrador o el área de Recursos Humanos (RRHH).

El proceso comienza cuando el usuario accede al módulo de ODI a través de la vista V\_Odi, donde se muestra la opción para crear una nueva orden. El usuario procede a agregar los datos necesarios (información del trabajador, cargo, fecha de ingreso, condiciones y observaciones), tras lo cual el sistema renderiza una previsualización del documento.

Posteriormente, el usuario presiona crear ODI, generando el modelo de la orden mediante el método query Generate PDF(), que es validado por C\_Pydantic antes de enviarse al controlador C\_create\_ODI. Este ejecuta la función sqlalchemy.generate-pdf, la cual realiza la instrucción ADD \* TO Odi en la base de datos, confirmando con el código de retorno 200.

Finalmente, el sistema genera y devuelve el archivo PDF de la Orden de Ingreso, que es enviado a la vista para su descarga directa por el usuario, completando el proceso con la notificación “Descargar PDF”.

### *Figura* 9.25 *“Diagrama de Secuencia - caso de uso 43”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.26 Caso de uso N°44**

La figura 9.26 representa el flujo de interacción para subir una Orden de Ingreso (ODI) asociada a un trabajador dentro del sistema. Este proceso es ejecutado por el Administrador o el área de Recursos Humanos (RRHH).

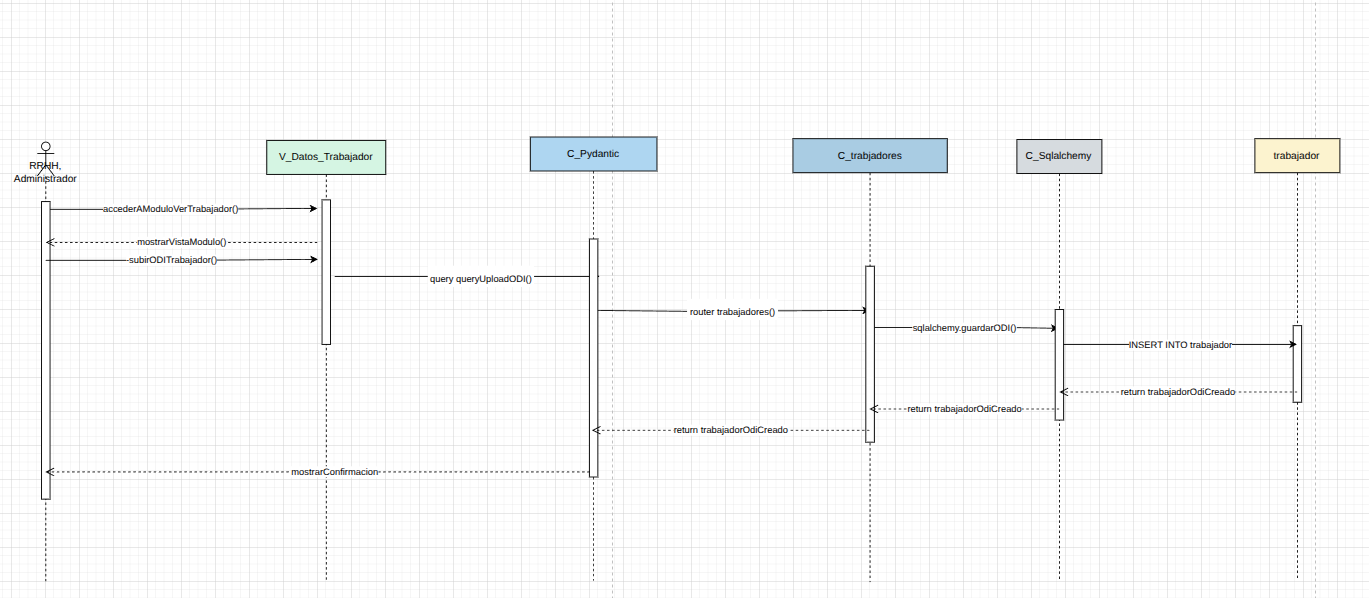
El flujo comienza cuando el usuario accede al módulo de visualización de trabajadores mediante la vista V\_Datos\_Trabajador, donde el sistema muestra la interfaz con la opción para subir ODI de trabajador.

Al ejecutar la acción subirODITrabajador(), la vista envía la solicitud queryUploadODI() al esquema C\_Pydantic para validar la estructura del documento y los datos asociados al trabajador.

Una vez validados, se invoca el controlador C\_trabajadores, que ejecuta el método sqlalchemy.guardarODI() encargado de realizar la operación INSERT INTO trabajador, registrando la orden dentro del perfil del trabajador en la base de datos.

Finalmente, el sistema devuelve la confirmación trabajadorOdiCreado, mostrando un mensaje de éxito en la vista, indicando que la Orden de Ingreso fue cargada correctamente y vinculada al expediente del trabajador.

### *Figura* 9.26 *“Diagrama de Secuencia - caso de uso 44”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.27 Caso de uso N°45**

La figura 9.27 detalla el flujo correspondiente a la generación y descarga del PDF de una Orden de Ingreso (ODI) ya registrada en el sistema. Este proceso puede ser ejecutado por el Administrador o por el área de Recursos Humanos (RRHH).

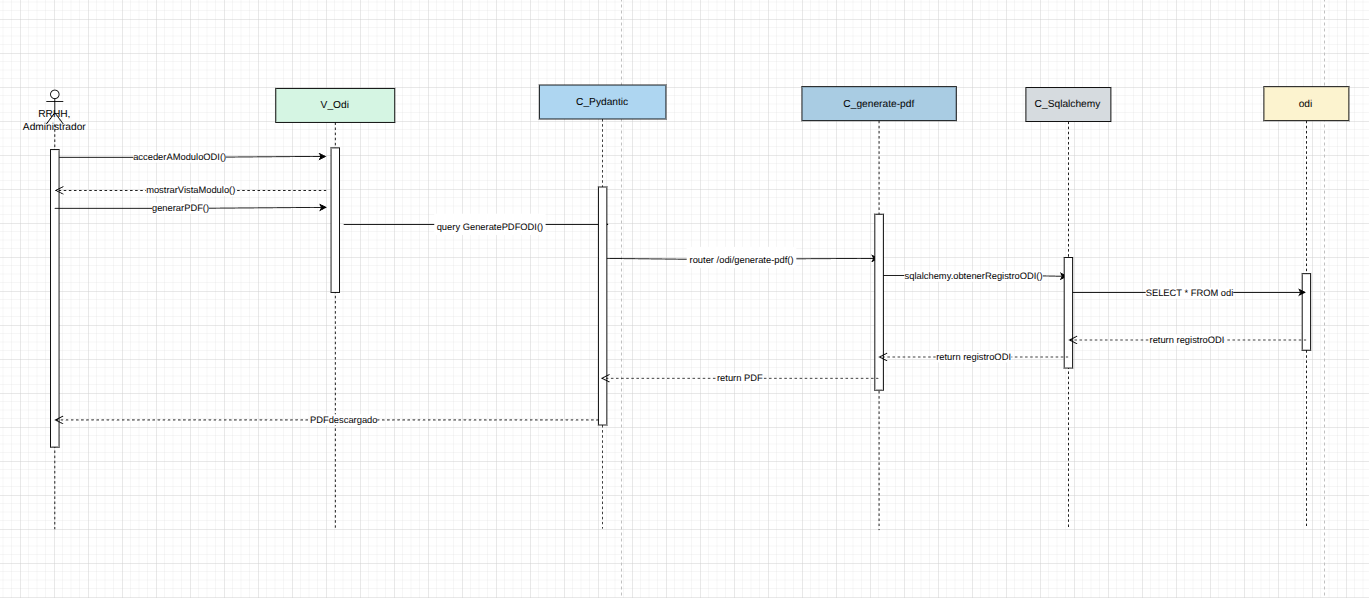
El proceso inicia cuando el usuario accede al módulo de ODI desde la vista V\_Odi, donde el sistema muestra la interfaz principal del módulo. A continuación, el usuario selecciona la opción “Generar PDF”, lo que desencadena la ejecución del método generarPDF().

La vista envía la solicitud query GeneratePDFODI() al esquema C\_Pydantic, el cual valida los datos del registro antes de redirigir la solicitud al controlador C\_generate-pdf.

Este controlador invoca el método sqlalchemy.obtenerRegistroODI(), que realiza una consulta SELECT \* FROM odi para obtener la información almacenada en la base de datos.

Una vez recuperados los datos, el sistema genera el documento en formato PDF, devolviendo el resultado al cliente. Finalmente, la vista muestra el mensaje PDFdescargado, indicando que el archivo ha sido generado y descargado exitosamente, completando así el proceso.

### *Figura* 9.27 *“Diagrama de Secuencia - caso de uso 45”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.28 Caso de uso N°46**

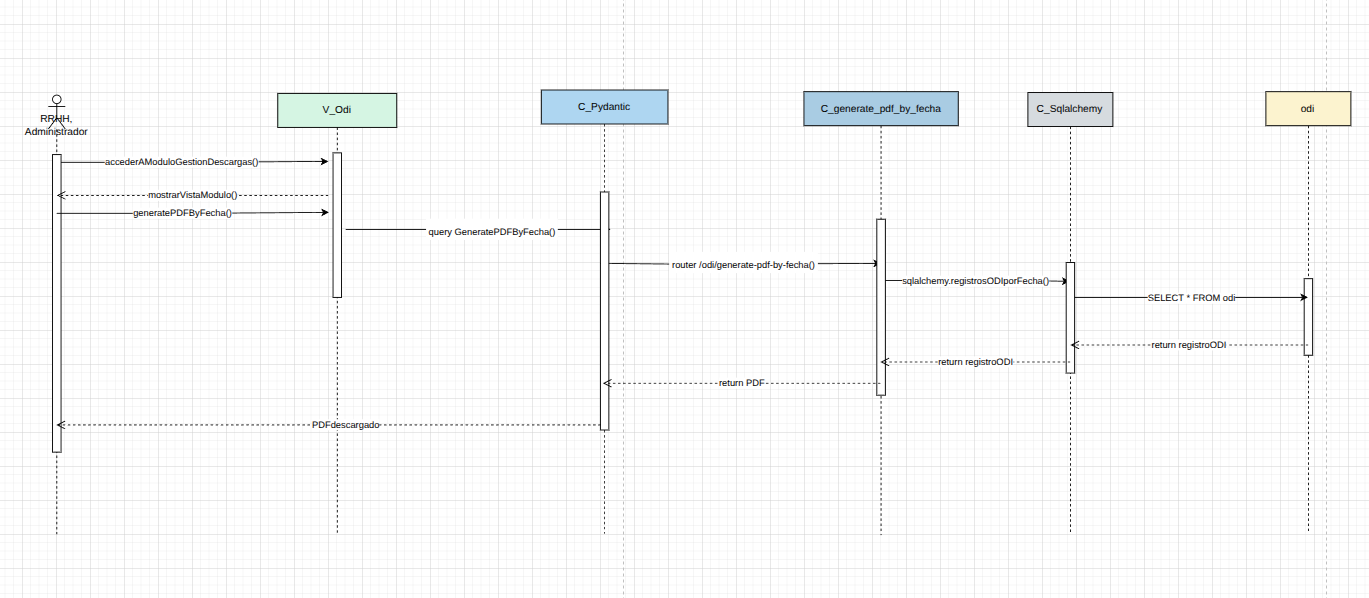
La figura 9.28 muestra el flujo de interacción correspondiente a la generación de reportes PDF de Órdenes de Ingreso (ODI) filtradas por fecha. Este proceso es iniciado por el Administrador o el área de Recursos Humanos (RRHH) a través del módulo de gestión de descargas.

El proceso comienza cuando el usuario accede al módulo mediante la vista V\_Odi, que presenta la interfaz con las opciones de filtrado temporal. Posteriormente, el usuario ejecuta la acción generatePDFByFecha(), solicitando la generación del reporte para un intervalo específico.

La vista envía la consulta query GeneratePDFByFecha() al esquema C\_Pydantic, que valida los parámetros de fecha y remite la solicitud al controlador C\_generate\_pdf\_by\_fecha. Este, a su vez, ejecuta el método sqlalchemy.registrosODIporFecha(), el cual realiza la instrucción SELECT \* FROM odi aplicando el filtro de fechas correspondiente para obtener los registros relevantes.

Una vez obtenidos los resultados, el sistema compila los datos y genera el archivo PDF consolidado, el cual es retornado a la vista junto con la notificación PDFdescargado, indicando que el reporte de ODIs filtradas por fecha fue generado y descargado exitosamente.

### *Figura* 9.28 *“Diagrama de Secuencia - caso de uso 46”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **9.29 Caso de uso N°47**

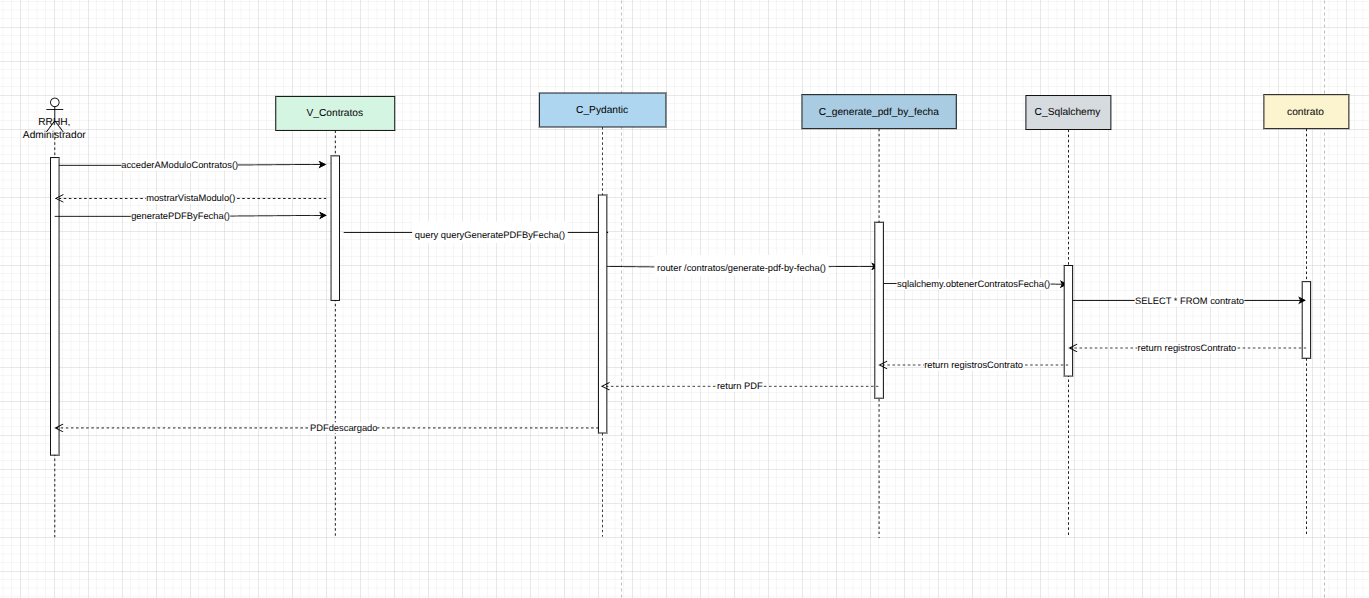
La figura 9.29 muestra el flujo correspondiente a la generación de reportes en PDF de contratos filtrados por rango de fechas, funcionalidad utilizada por el Administrador o el área de Recursos Humanos (RRHH) para obtener documentos consolidados de los contratos emitidos en un periodo determinado.

El proceso comienza cuando el usuario accede al módulo de contratos mediante la vista V\_Contratos, donde el sistema presenta las opciones de filtrado temporal. Al ejecutar la acción generatePDFByFecha(), la vista envía la solicitud query GeneratePDFByFecha() al esquema C\_Pydantic, encargado de validar los parámetros de búsqueda (fechas de inicio y fin).

Luego, se redirige la petición al controlador C\_generate\_pdf\_by\_fecha, el cual invoca el método sqlalchemy.obtenerContratosFecha() para realizar la consulta SELECT \* FROM contrato, filtrando los registros dentro del rango temporal definido.

Una vez obtenidos los datos, el sistema genera el reporte PDF de contratos, lo retorna al cliente y muestra la notificación PDFdescargado, indicando que el archivo fue generado y descargado correctamente, cumpliendo con la función de respaldo documental y análisis histórico del módulo de contratos.

### *Figura* 9.29 *“Diagrama de Secuencia - caso de uso 47”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

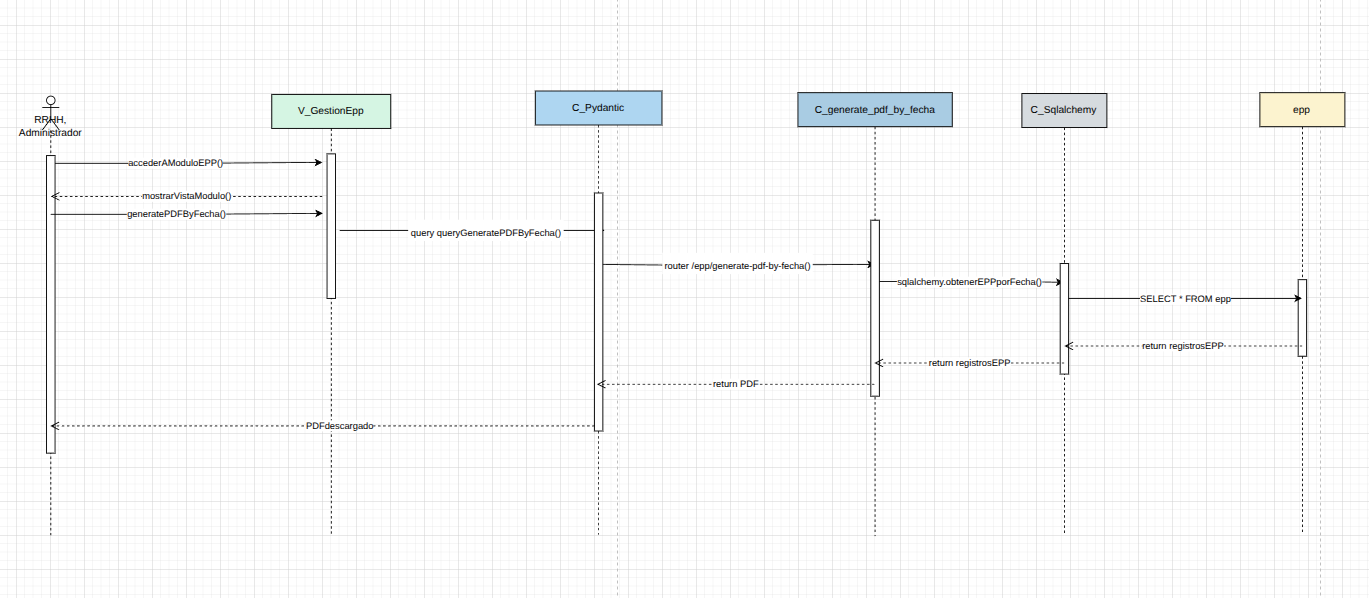
## **9.30 Caso de uso N°48**

La figura 9.30 muestra el flujo de generación de reportes PDF de entregas de Elementos de Protección Personal (EPP) filtrados por fechas, utilizado por el Administrador o RRHH para control documental.

El proceso inicia cuando el usuario accede al módulo de gestión de EPP desde la vista V\_GestionEpp y ejecuta generatePDFByFecha(). La solicitud es validada por C\_Pydantic y enviada al controlador C\_generate\_pdf\_by\_fecha, que ejecuta SELECT \* FROM epp mediante sqlalchemy.obtenerEPPporFecha() para obtener los registros del periodo.

Finalmente, el sistema genera y devuelve el PDF consolidado, mostrando la confirmación PDFdescargado, indicando que el reporte fue generado correctamente.

### *Figura* 9.30 *“Diagrama de Secuencia - caso de uso 48”*



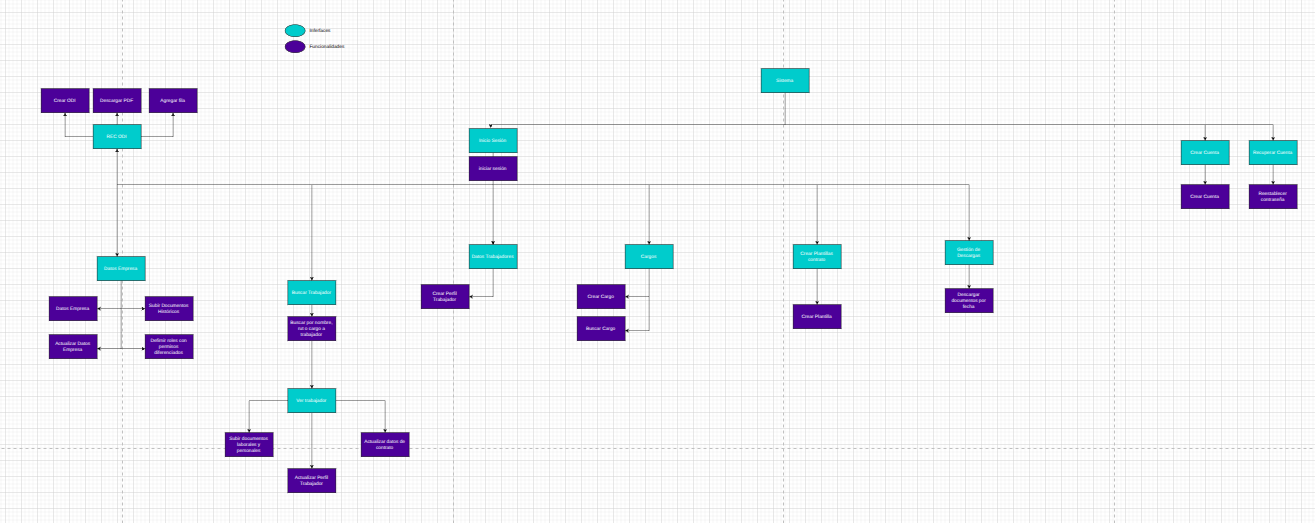
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# 

# **10. ÁRBOL DE NAVEGACIÓN**

En este apartado se presenta la trazabilidad entre el árbol de navegación de la interfaz y los casos de uso definidos para el Avance Scrum++ 2. El objetivo es mostrar cómo cada elemento del menú y sus subniveles se corresponden con los casos de uso críticos, garantizando que la estructura de navegación soporte de forma clara y directa las funcionalidades esenciales de ContaPlus. De este modo, se facilita tanto el diseño de la interfaz como la validación de los flujos de usuario, alineando la experiencia de navegación con los objetivos del sistema.

### *Figura* 10.1 *“Diagrama de navegación”*



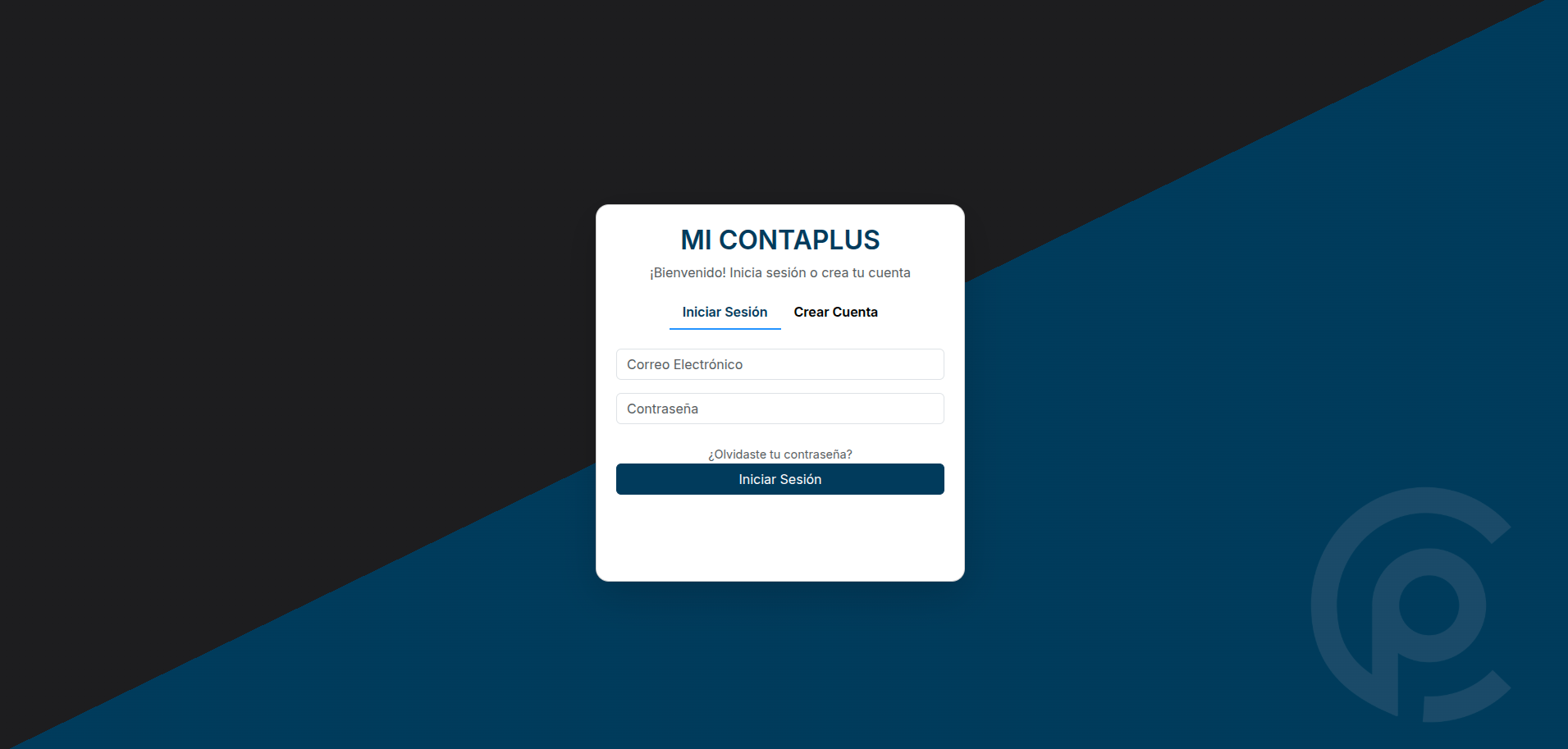
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## **10.1 Vista “Sistema”**

Corresponde a la vista principal o padre del sistema, desde la cual se desprenden las demás vistas como Crear cuenta, Recuperar contraseña e Ingresar. Esta vista actúa como contenedor general de la aplicación, gestionando la estructura y navegación inicial. Al no poseer una lógica de negocio propia, no requiere una capa de controladores, limitándose a manejar la lógica visual y de enrutamiento entre las vistas secundarias.

### *Figura 10.2 “Vista “Sistema”*

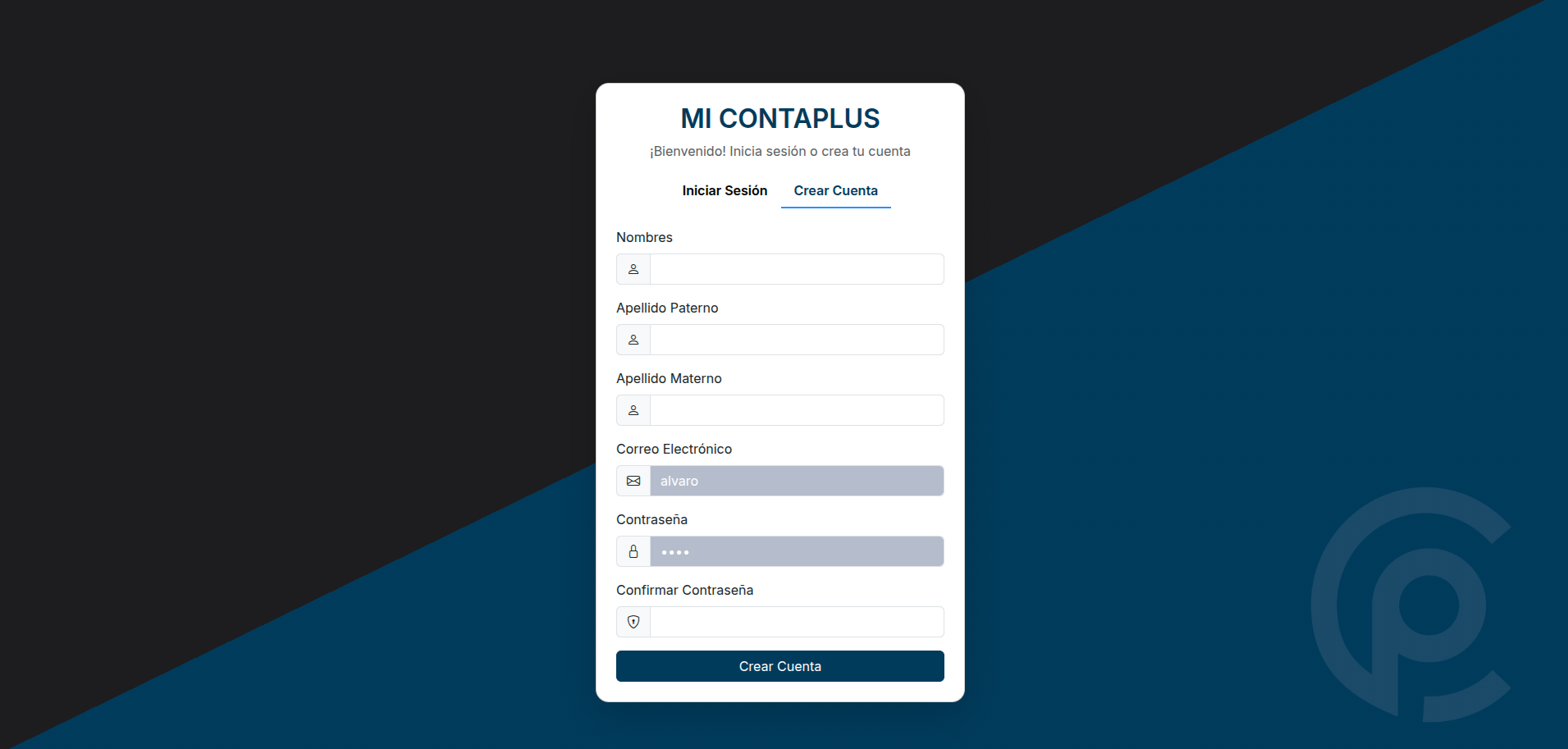


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.2 Vista “Crear Cuenta”**

Corresponde a la vista usada para la creación de las cuentas, en ella se encontrarán datos como correo, contraseña, nombre, apellido paterno, apellido materno. Solo tiene la función “Crear cuenta” que envía los datos a la capa de controladores para poder validar y guardar los datos.

### *Figura 10.3 “Crear Cuenta”*

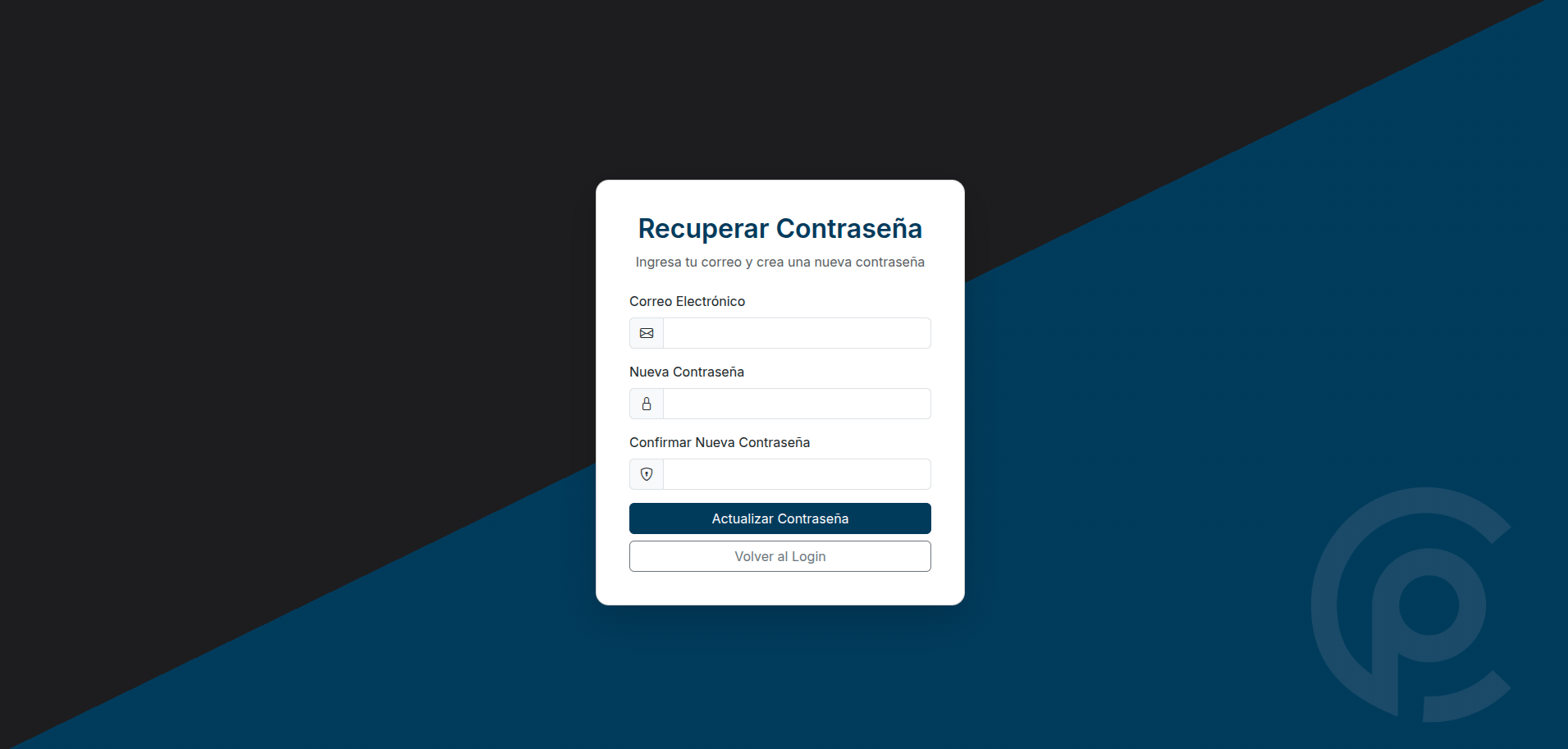


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.3 Vista “Recuperar Cuenta”**

Vista diseñada para que el actor recupere su contraseña, solicita correo y nueva contraseña para luego enviar esta solicitud a la capa controladores la cual enviará un correo al actor para validar esta operación

### *Figura 10.4 “Recuperar Cuenta*

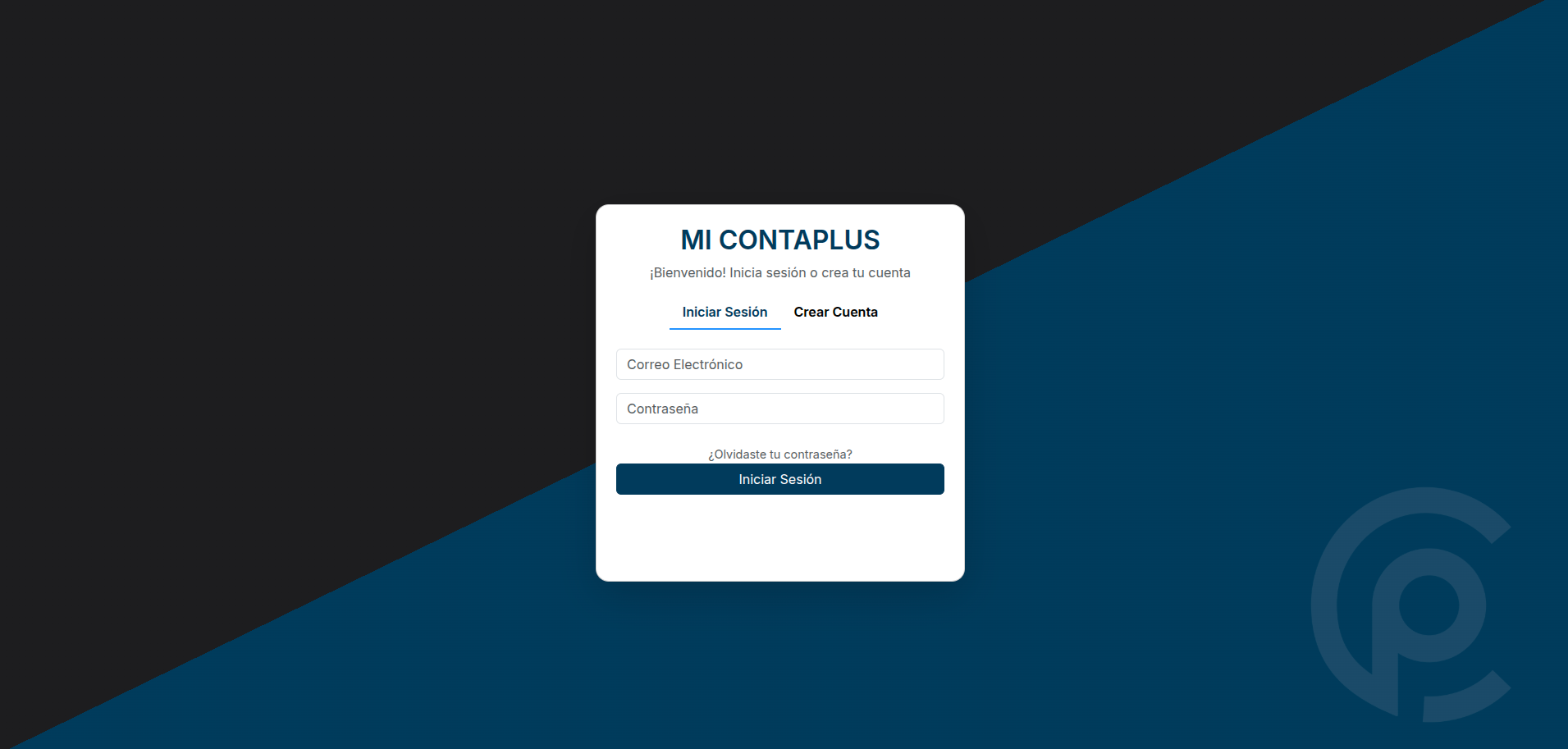


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**10.4 Vista “Inicio Sesión”**

Vista que permite al actor iniciar sesión mediante usuario y contraseña, la vista mantiene una función llamada “iniciar sesión” la cual envía los datos a la capa controladores

### *Figura 10.5 “Iniciar Sesión”*

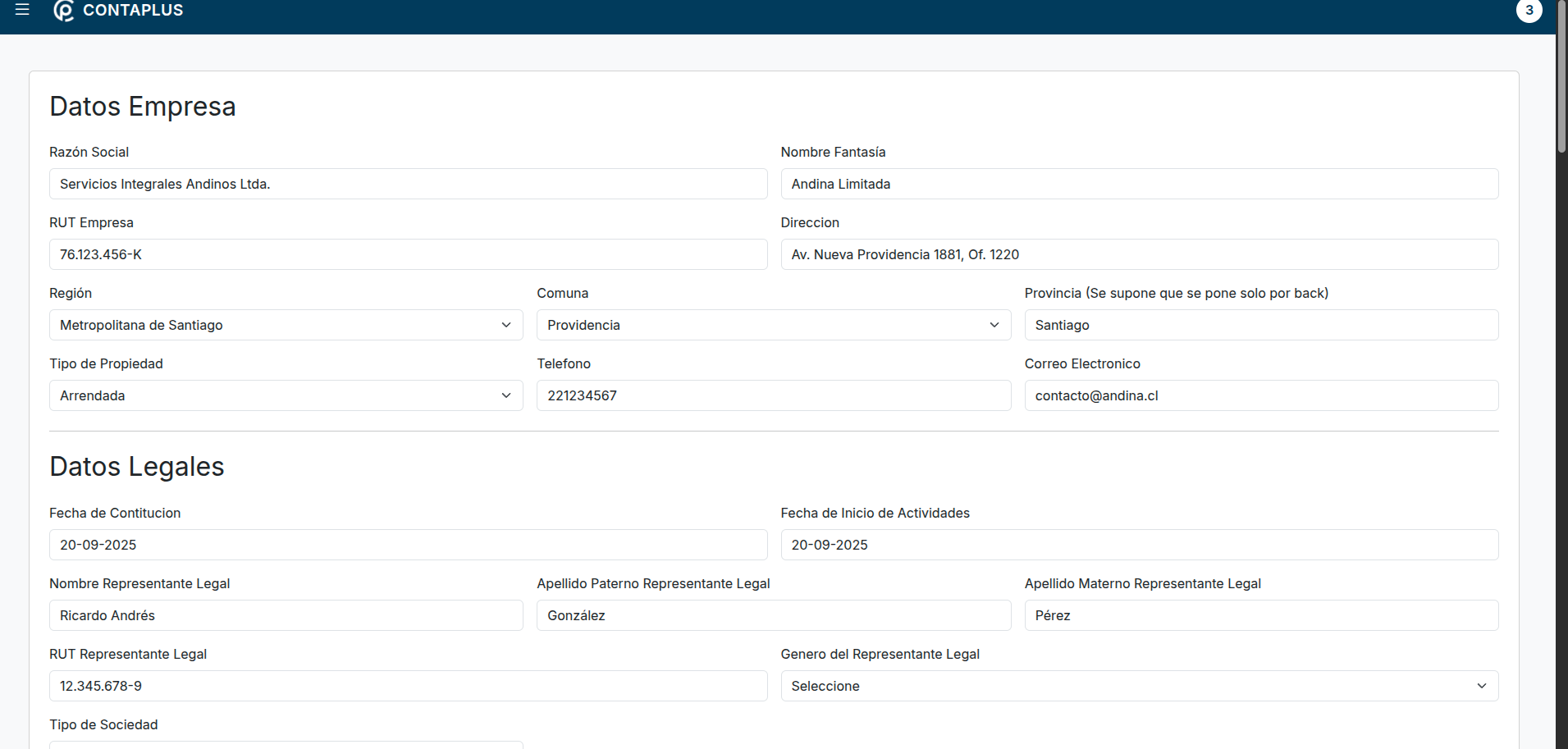


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.5 Vista “Datos Empresa”**

Vista que permite ingresar datos de socios, editarlos y eliminarlos. Para ingresar a esta vista se necesita tener rol de administrador, tener el correo validado e iniciar sesión. La vista contiene cuatro funciones que sirven para actualizar los datos, ingresar datos, definir roles y subir documentos.

### *Figura 10.6 “Datos Empresa”*

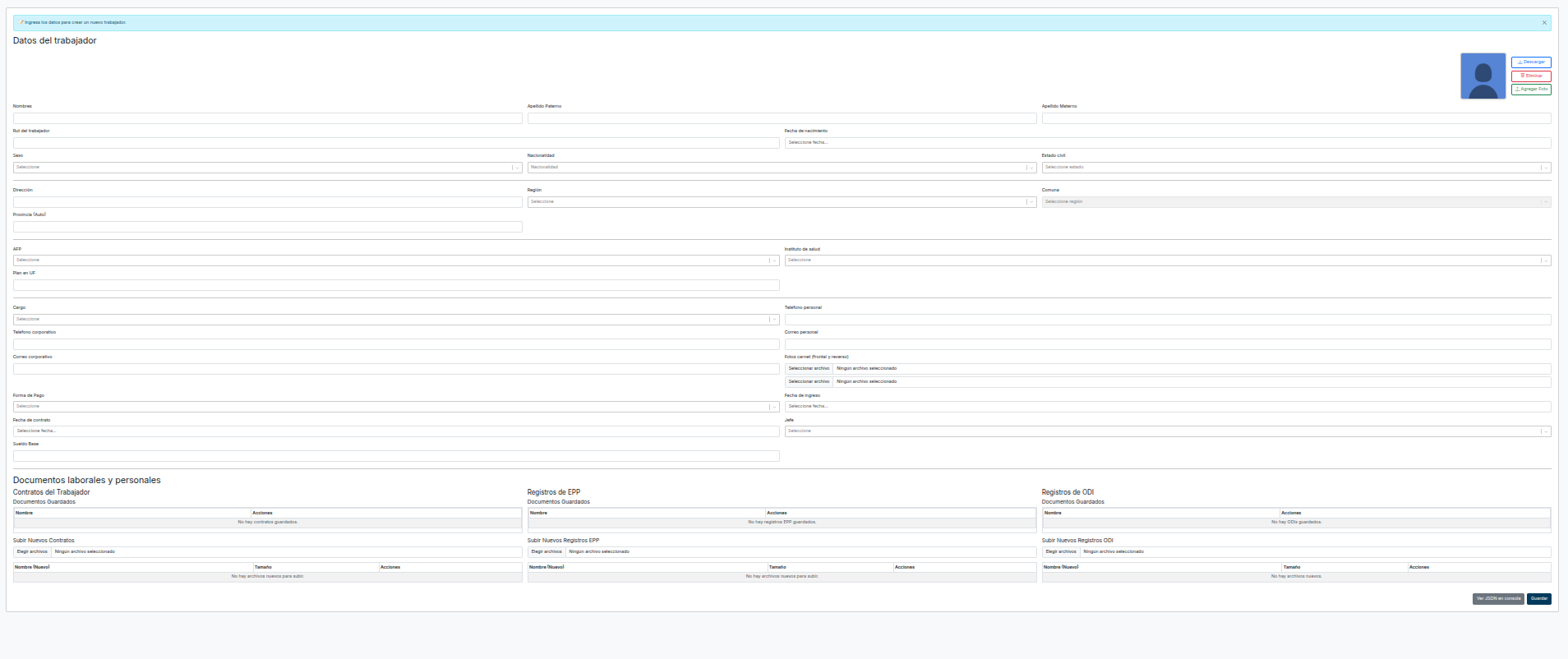
****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.6 Vista “Crear Trabajadores”**

Vista diseñada para crear un nuevo perfil del trabajador, en donde deberá colocar datos como rut, nombre, teléfono de contacto, etc.

### *Figura 10.7 “Crear Trabajadores”*

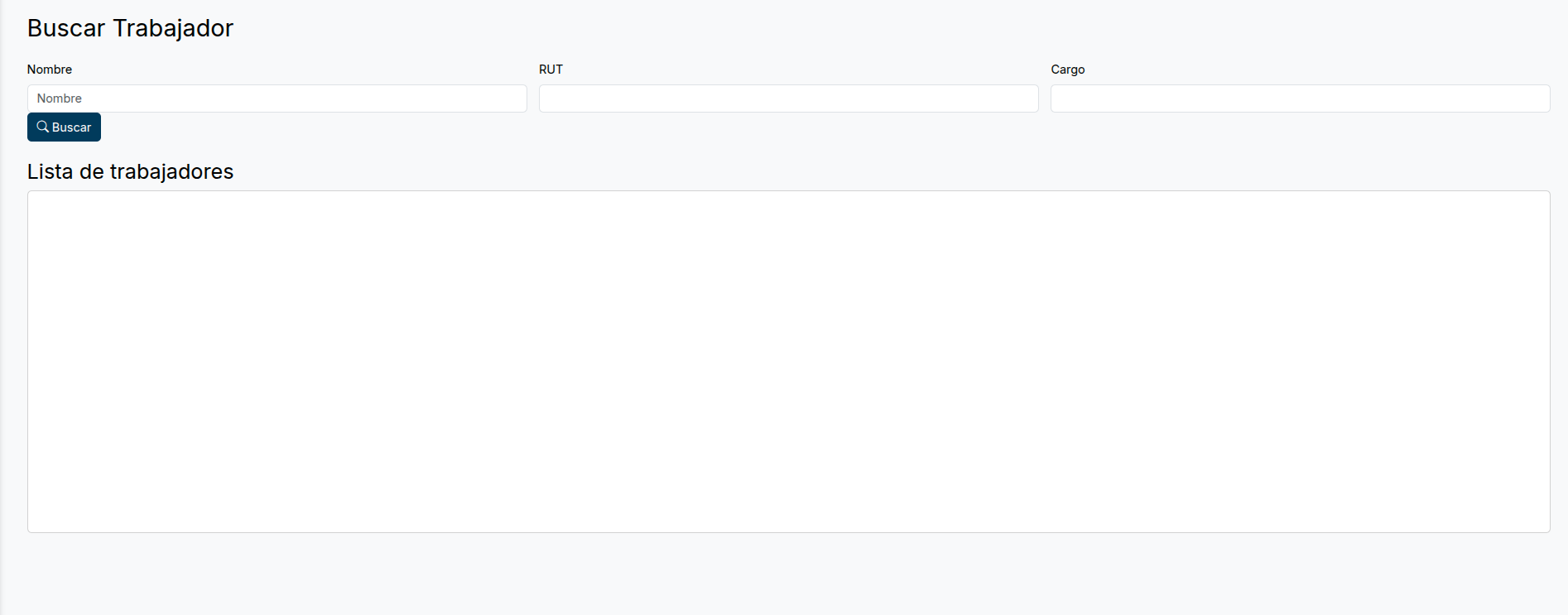


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.6 Vista “Buscar Trabajador”**

Vista creada para poder buscar un trabajador gracias a su función que permite buscar a uno o varios trabajadores por rut, cargo, nombre. Pudiendo así seleccionar al buscado.

### *Figura 10.8 “Buscar Trabajador”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## **10.7 Vista “Ver Trabajador”**

Vista desarrollada para poder visualizar y editar la información de un trabajador, para poder acceder a esta vista se tiene que buscar y seleccionar un trabajador en la vista de “Buscar Trabajador”. Esta vista contiene funciones que permiten actualizar, eliminar información del trabajador, además de poder subir documentos varios dentro de su perfil

### *Figura 10.8 “Buscar Trabajador”*

# 

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.8 Vista “Crear Plantillas de contrato”**

Vista creada para poder crear plantillas con variables dinámicas gracias a su botonera y la función “Crear plantilla”.

### *Figura 10.9 “Buscar Trabajador”*



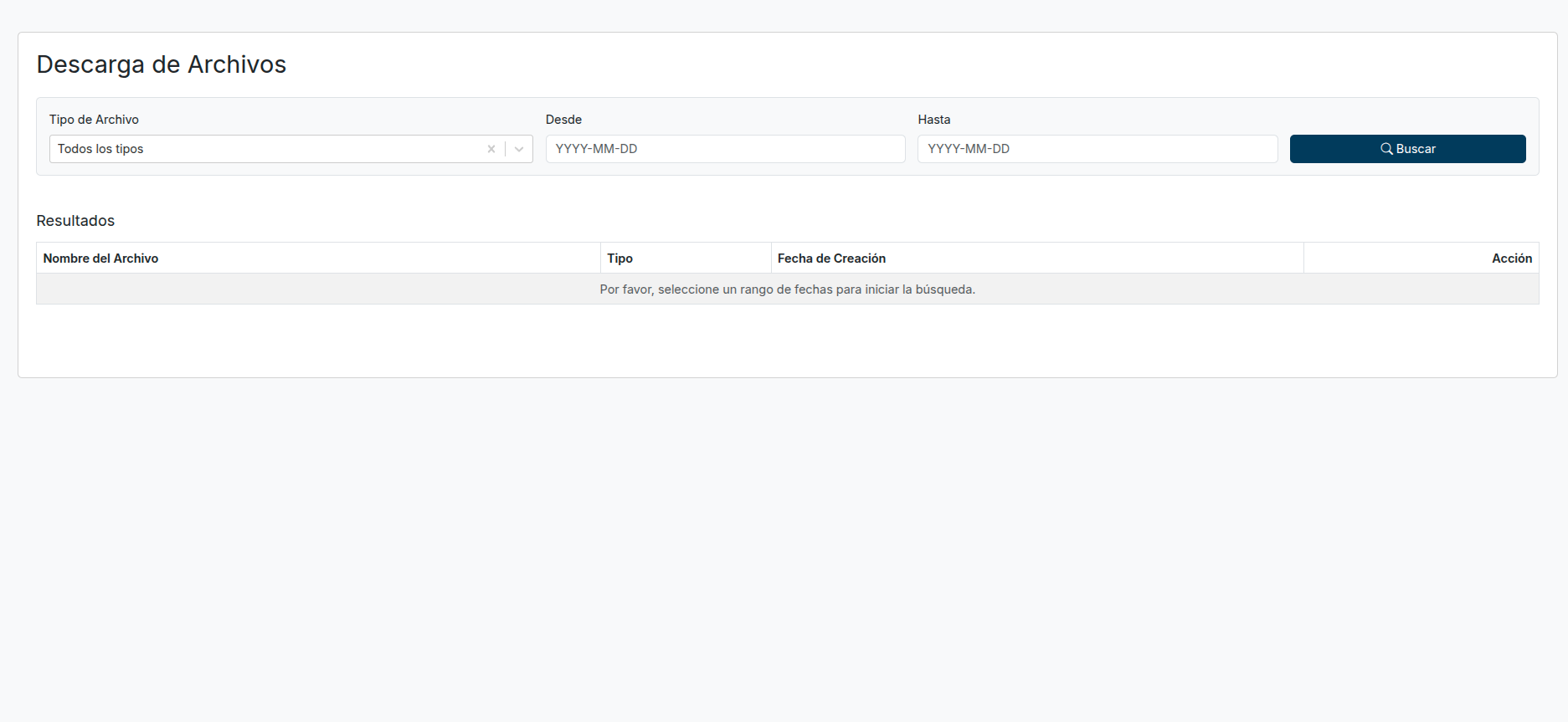
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## 

## **10.9 Vista “Gestión de Descargas”**

Vista creada para descargar los archivos generales, esto permite filtrar por tipo de archivo (contrato, E.P.P, O.D.I) y filtrar con fecha.

### *Figura 10.10 “Gestión de descargas”*

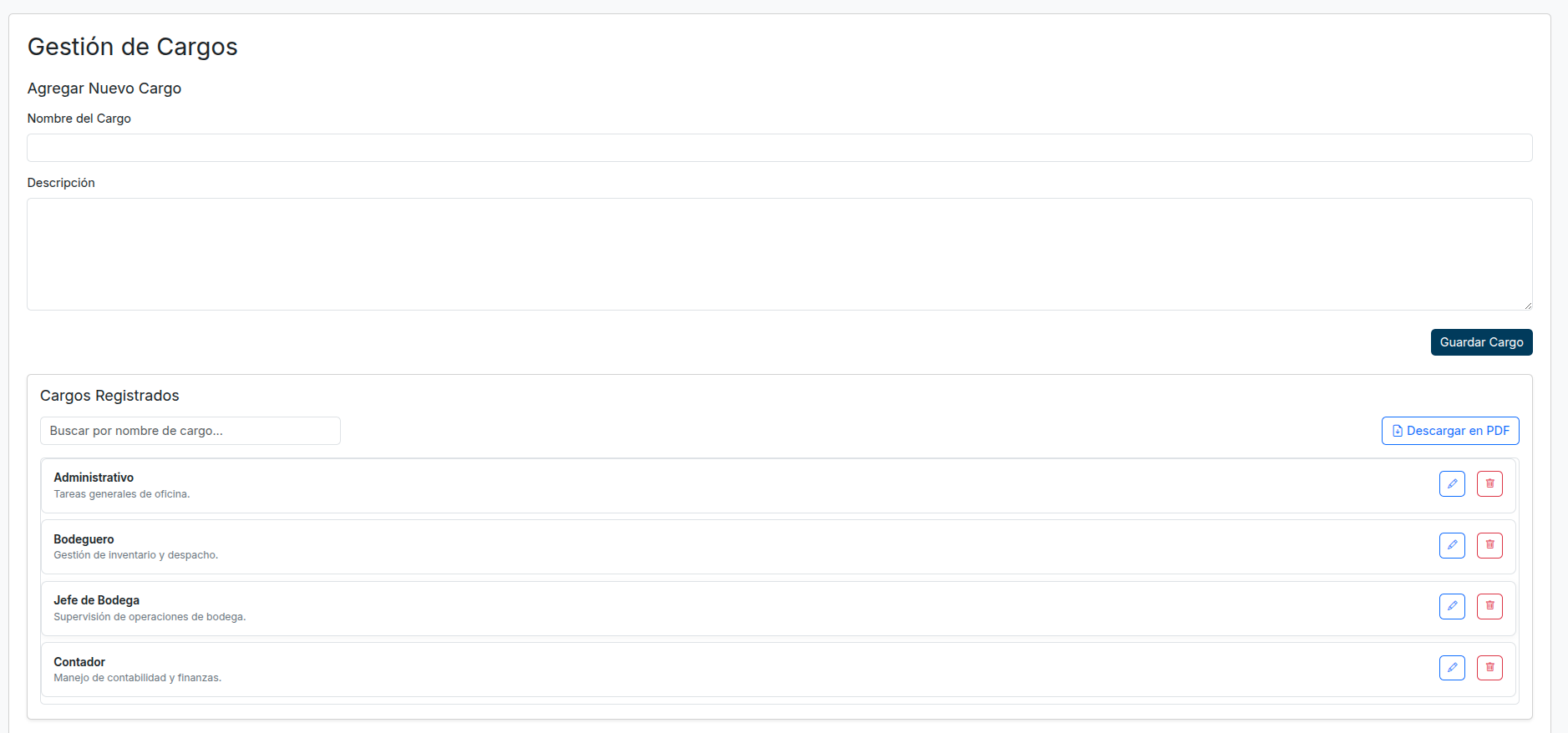


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.10 Vista “Cargos”**

Vista creada para poder gestionar los cargos en general, esta vista tiene dos funciones principales, “Crear cargo” y “Buscar cargo”, funciones que se pueden ver en la *Figura 10.11 a continuación.*

### *Figura 10.11 “Gestión de descargas”*

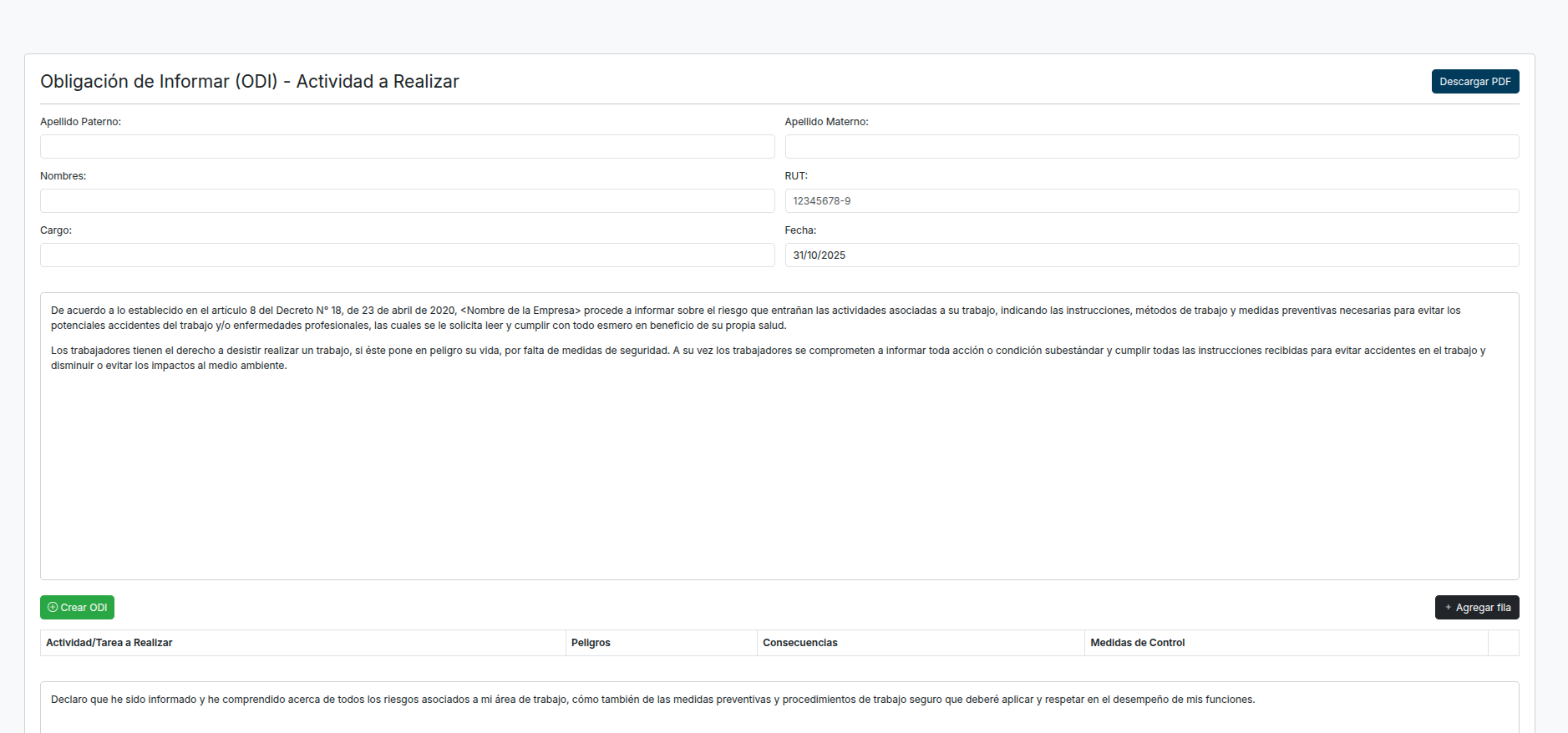


Fuente: Elaborado por el estudiante de acuerdo con el proyecto

## **10.11 Vista “REC ODI”**

Vista creada para poder generar y descargar un ODI gracias a tres funciones principales: Crear ODI, Descargar PDF, Agregar fila, a continuación en la Figura 10.12 se puede ver la vista de esto.

### *Figura 10.12 “REC ODI”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# **11. PRUEBAS**

En este apartado se presentan las pruebas asociadas a cada caso de uso del Avance Scrum ++ 2. Su objetivo es validar que ContaPlus ejecute correctamente los flujos definidos, comprobando con datos de entrada y resultados esperados que cada funcionalidad cumpla los requisitos del usuario.

**Caso de Uso N°11:** agregando método de pago a socio

**Casos de Prueba:**

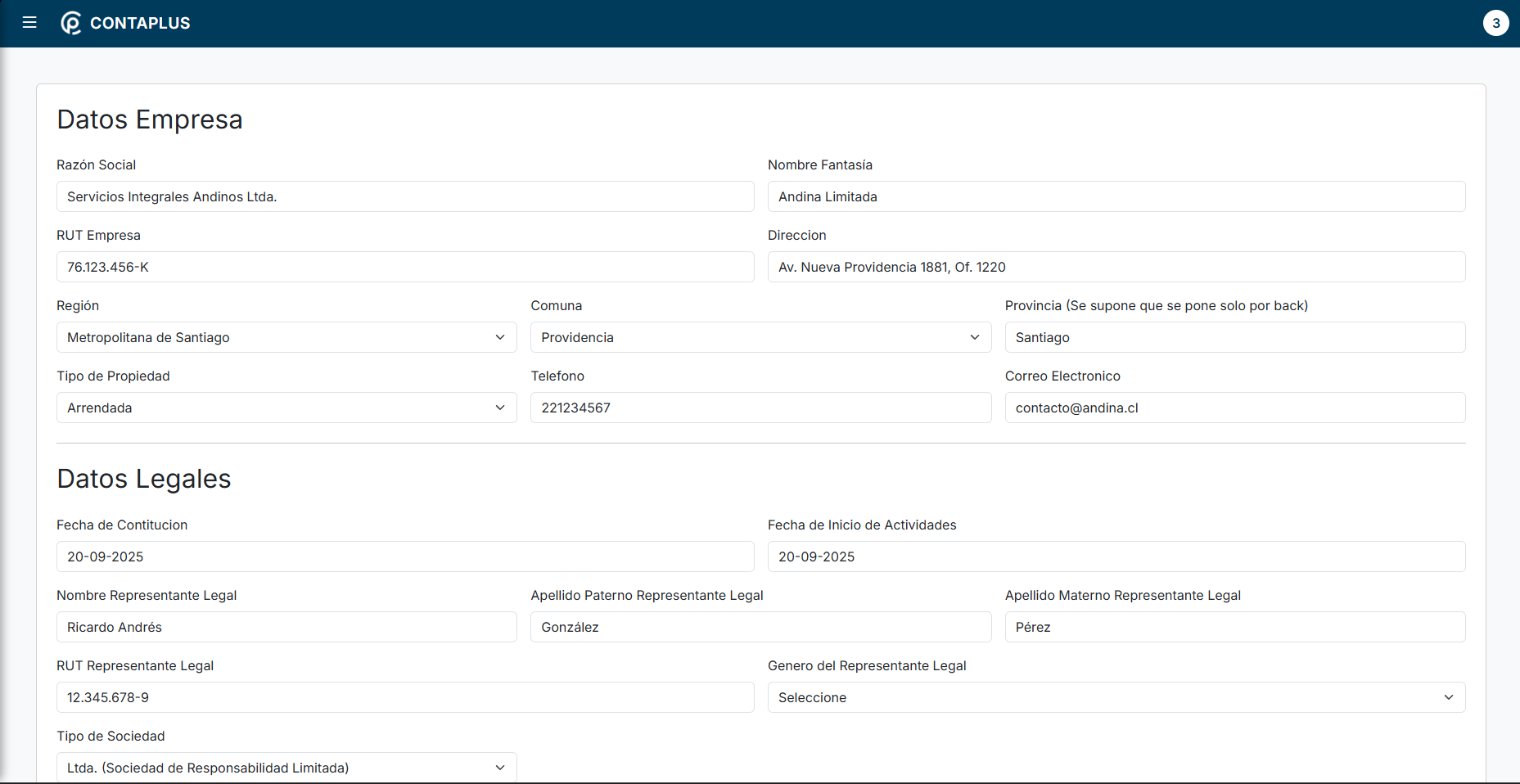
1. Ingresar al módulo de datos-empresa
2. pulsar el botón de método de pago
3. selecciona el método de pago
4. baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se renderiza una lista con las opciones de método de pago para el socio
3. Se guarda la opción seleccionada
4. Se guarda los cambios realizados

**Resultados Obtenidos:**

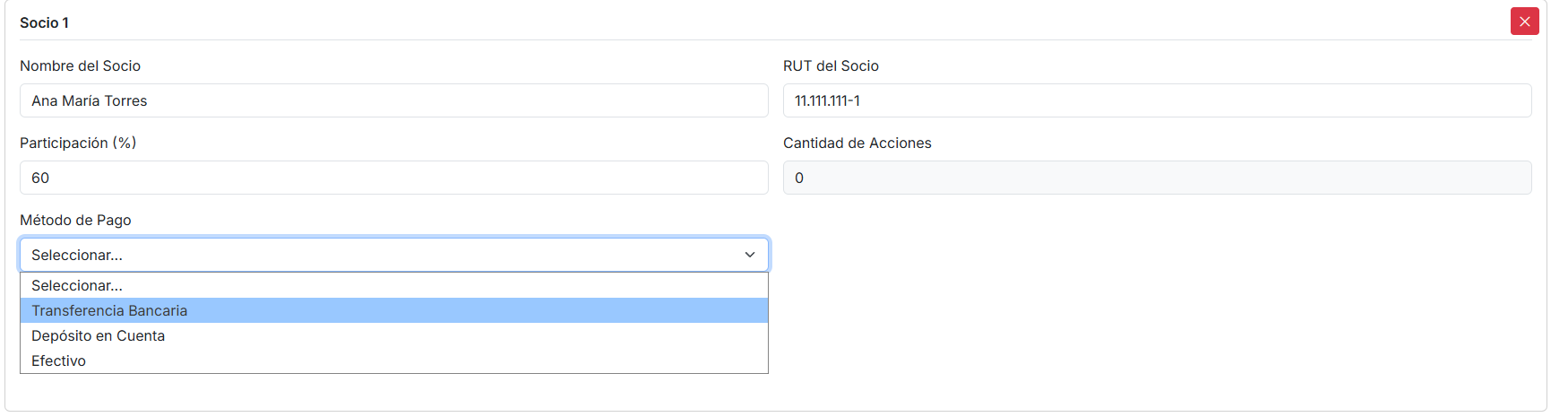
### *Figura* 11.1 *“Pruebas del sistema - Caso de uso 11”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### 

### *Figura* 11.1 *“Pruebas del sistema - Caso de uso 11”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.1 *“Pruebas del sistema - Caso de uso 11”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°12:** editando método de pago de socio

**Casos de Prueba:**

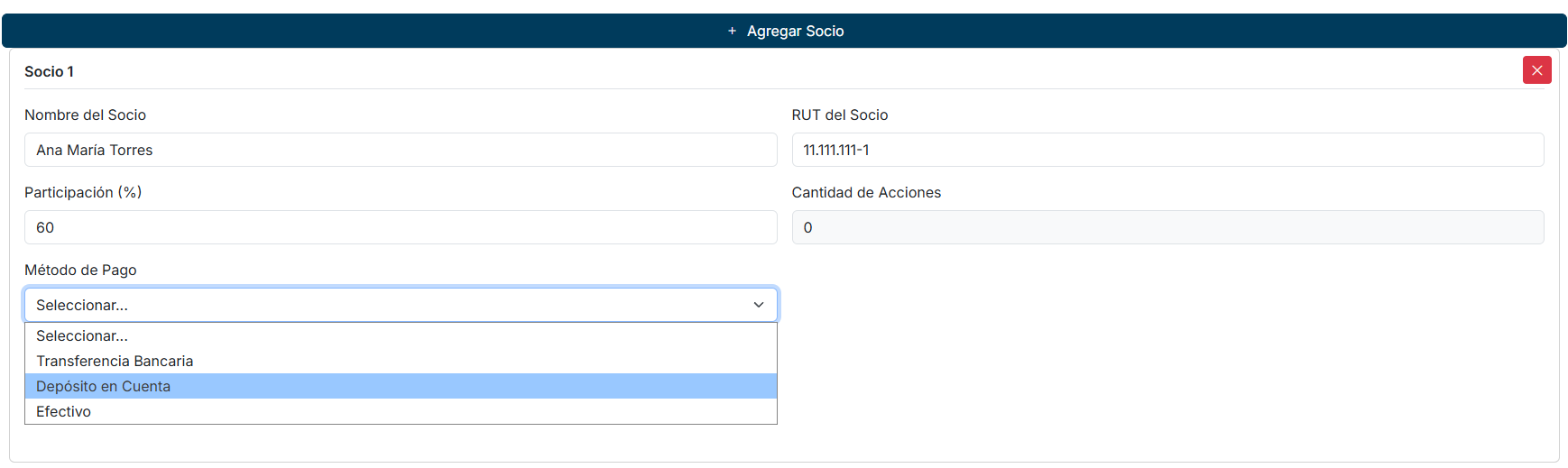
1. Ingresar al módulo de datos-empresa
2. pulsar el botón de método de pago
3. selecciona el método de pago a cambiar
4. baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se renderiza una lista con las opciones de método de pago para el socio
3. Se guarda la nuevo método de pago
4. Se guarda los cambios realizados

**Resultados obtenidos**

### *Figura* 11.2 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°13:** eliminando método de pago de socio

**Casos de Prueba:**

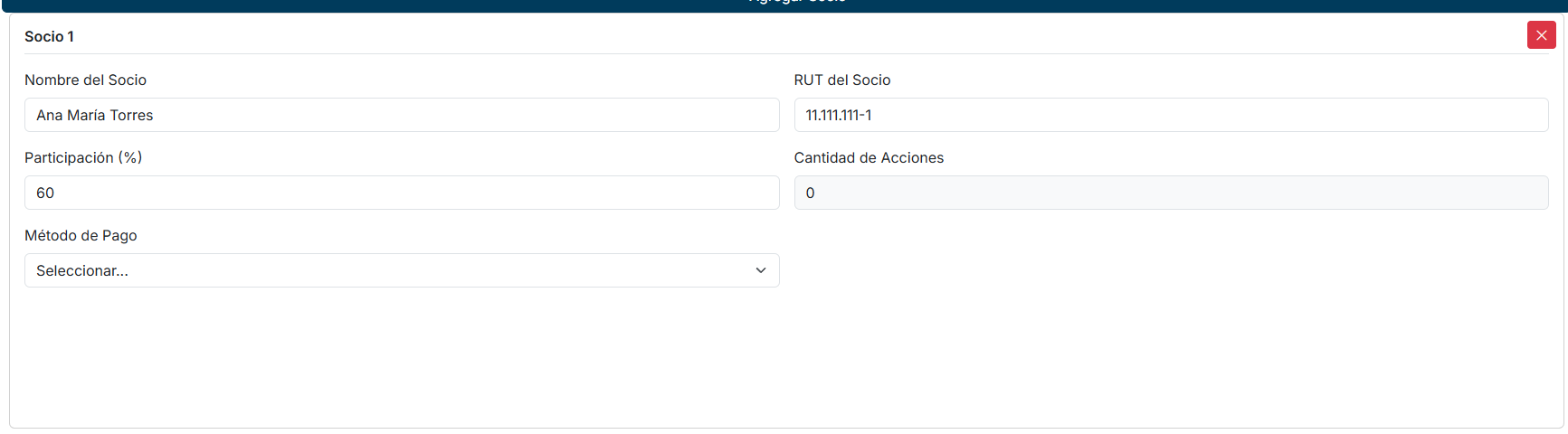
1. Ingresar al módulo de datos-empresa
2. pulsar el botón de método de pago
3. Selecciona el método de pago a “Seleccionar…”
4. baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se renderiza una lista con las opciones de método de pago para el socio
3. Se Seleccionar correctamente la opción default “Seleccionar”
4. Se guarda los cambios realizados

**Resultados obtenidos:**

### *Figura* 11.3 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°14:** editando acciones y capital total de la empresa

**Casos de Prueba:**

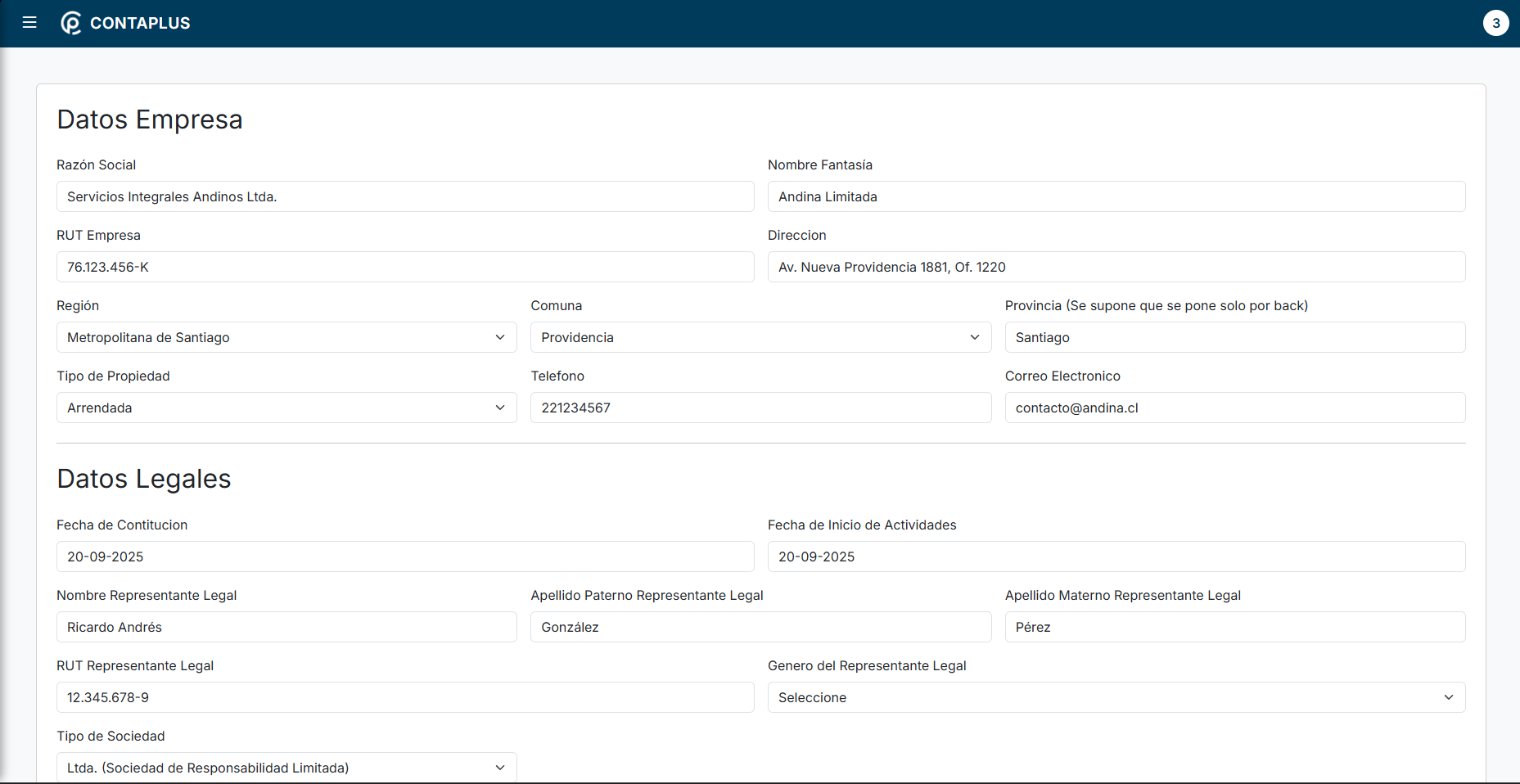
1. Ingresar al módulo de datos-empresa
2. Ingresar la cantidad de Acciones total
3. ingresar la el capital total
4. baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

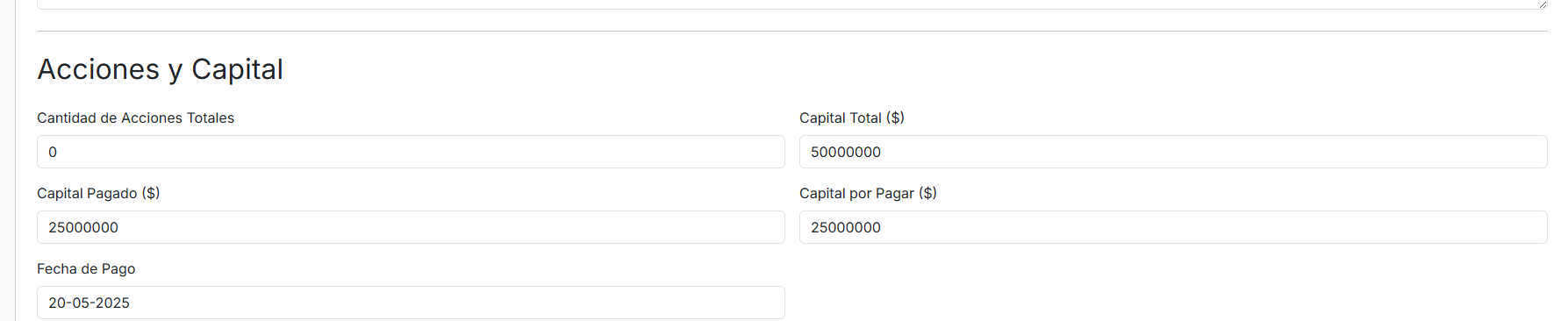
1. El módulo de Datos-Empresa se renderiza correctamente
2. Se queda el numero ingresado de Cantidad de Acciones Totales
3. Se queda el numero en Capital total
4. Se guarda los cambios realizados

**Resultado obtenidos**

### *Figura* 11.4 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

*Figura* 11.5 *“Pruebas del sistema”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°15:** editando direcciones de trabajo de la empresa

**Casos de Prueba:**

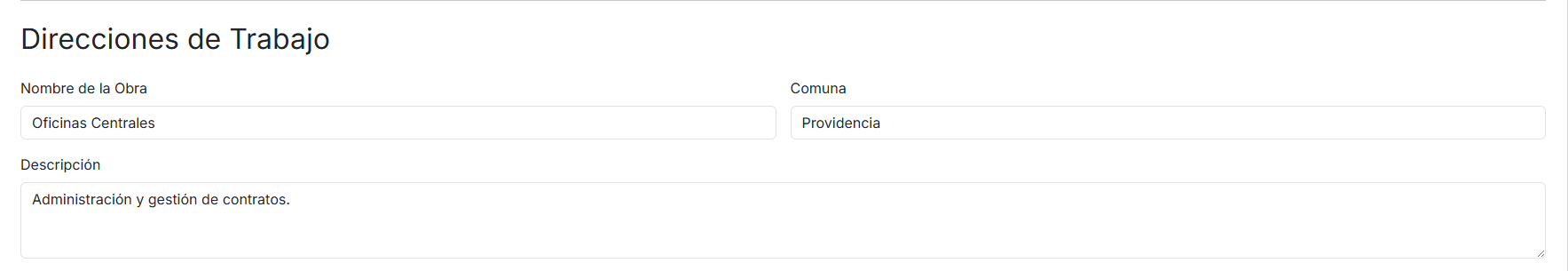
1. Ingresar al módulo de datos-empresa
2. Editar la comuna
3. baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se cambia la comuna
3. Se guarda los cambios realizados

**Resultados Obtenidos**

### *Figura* 11.5 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°16:** editando información de seguridad y previsión de la empresa

**Casos de Prueba:**

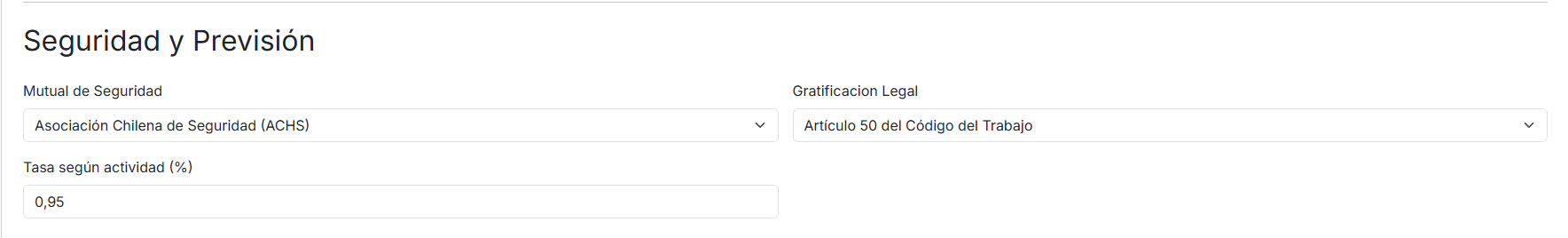
1. Ingresar al módulo de datos-empresa
2. Seleccionar la nueva mutual de Seguridad
3. Seleccionar el nuevo Gratificante Legal
4. Baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se renderiza la lista de las Mutuales de Seguridad para seleccionar uno
3. Se renderiza la lista de los Gratificantes Legales para seleccionar uno
4. Se guarda los cambios realizados

Resultados Obtenidos

### *Figura* 11.6 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°17:** editando actividad económica y régimen tributario de la empresa

**Casos de Prueba:**

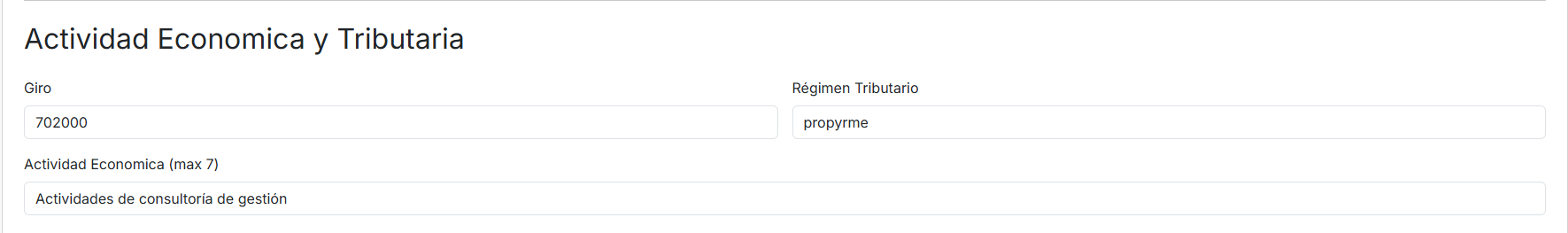
1. Ingresar al módulo de datos-empresa
2. Editar el número del Giro
3. Editar el Regimen Tributario
4. Editar la actividad Económica
5. baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. El número de giro se edita correctamente
3. El nombre del régimen tributario se edita correctamente
4. Las actividades Económicas se editan correctamente

Resultados Obtenidos

### *Figura* 11.7 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°18:** editando datos legales de la empresa

**Casos de Prueba:**

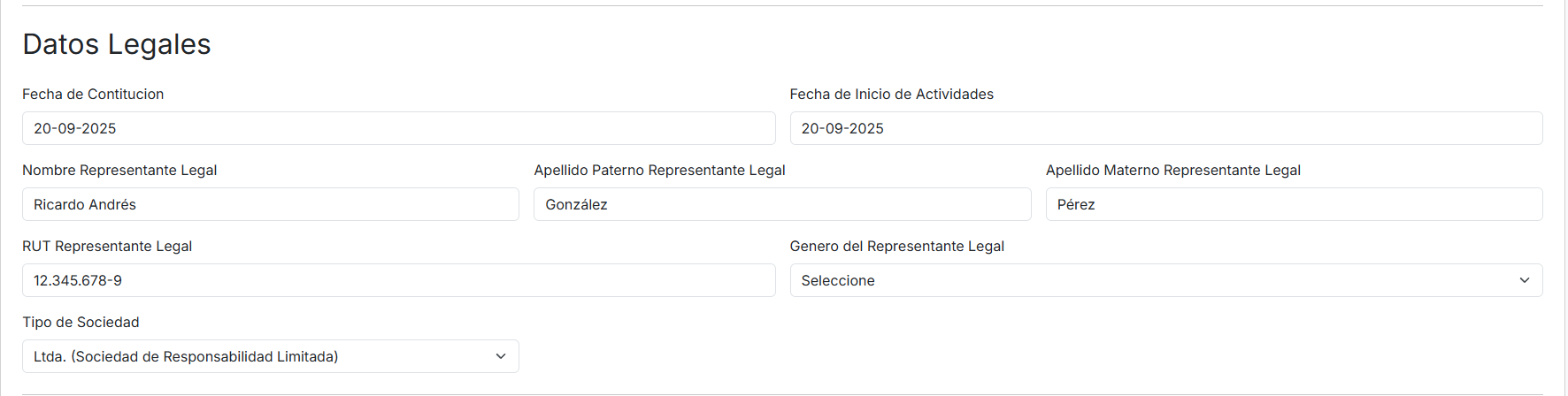
1. Ingresar al módulo de datos-empresa
2. Editar los datos dentro de la sección de Datos Legales
3. Baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se editan correctamente todos los datos de Datos legales de la empresa
3. Se guardan los cambios

**Resultados Obtenidos**

### *Figura* 11.8 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°19:** editando datos generales de la empresa

**Casos de Prueba:**

1. Ingresar al módulo de datos-empresa
2. Editar los datos dentro de la sección de Datos Empresa
3. Baja al final de la página y pulsa Guardar Datos

**Resultados Esperados:**

1. El módulo de Datos-Empresa se renderiza correctamente
2. Se edita correctamente los datos generales de la empresa
3. Se guardan los cambios

**Resultados Obtenidos**

### *Figura* 11.9 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°20:** creando perfil trabajador

**Casos de Prueba:**

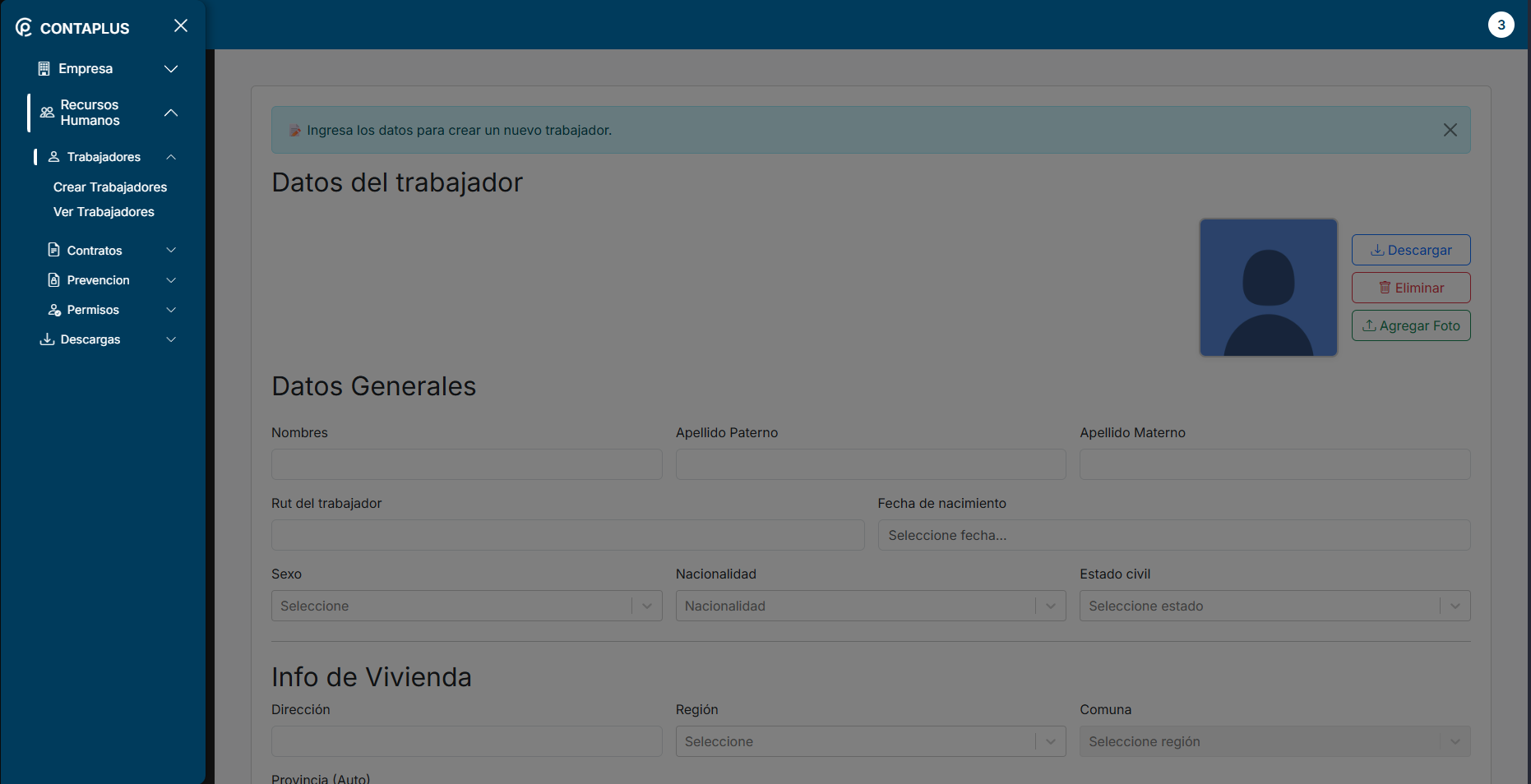
1. Ingresar al módulo de Crear Trabajadores
2. Rellenar los datos del trabajador
3. Ingresar Imagen del trabajador
4. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de Datos-trabajador se renderiza correctamente
2. Se rellenan correctamente los datos ingresados del trabajador
3. La imagen del trabajador se cambia por la nueva imagen subida
4. Se guardan los datos del trabajador

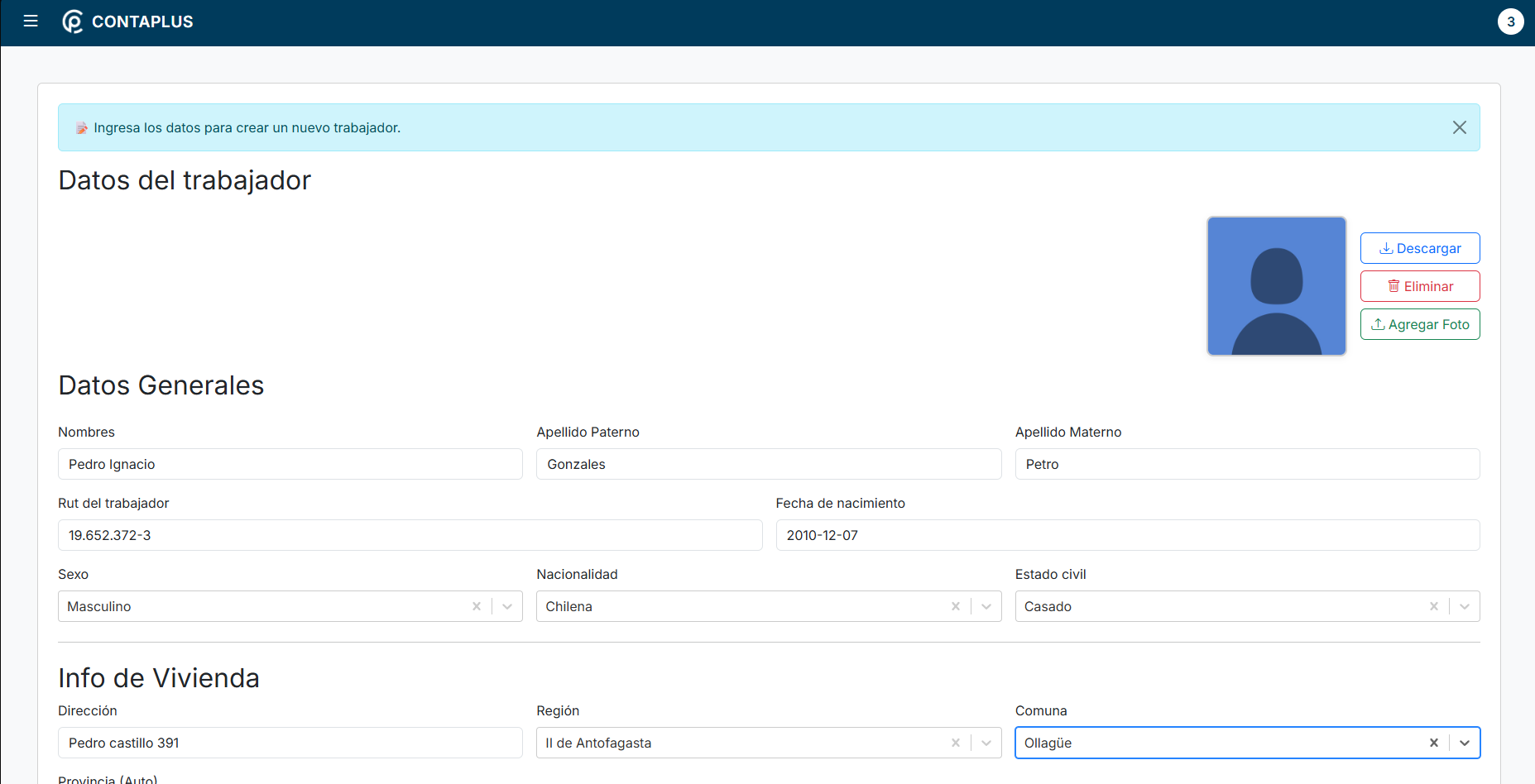
**Resultados Obtenidos**

### *Figura* 11.10 *“Pruebas del sistema”*

****

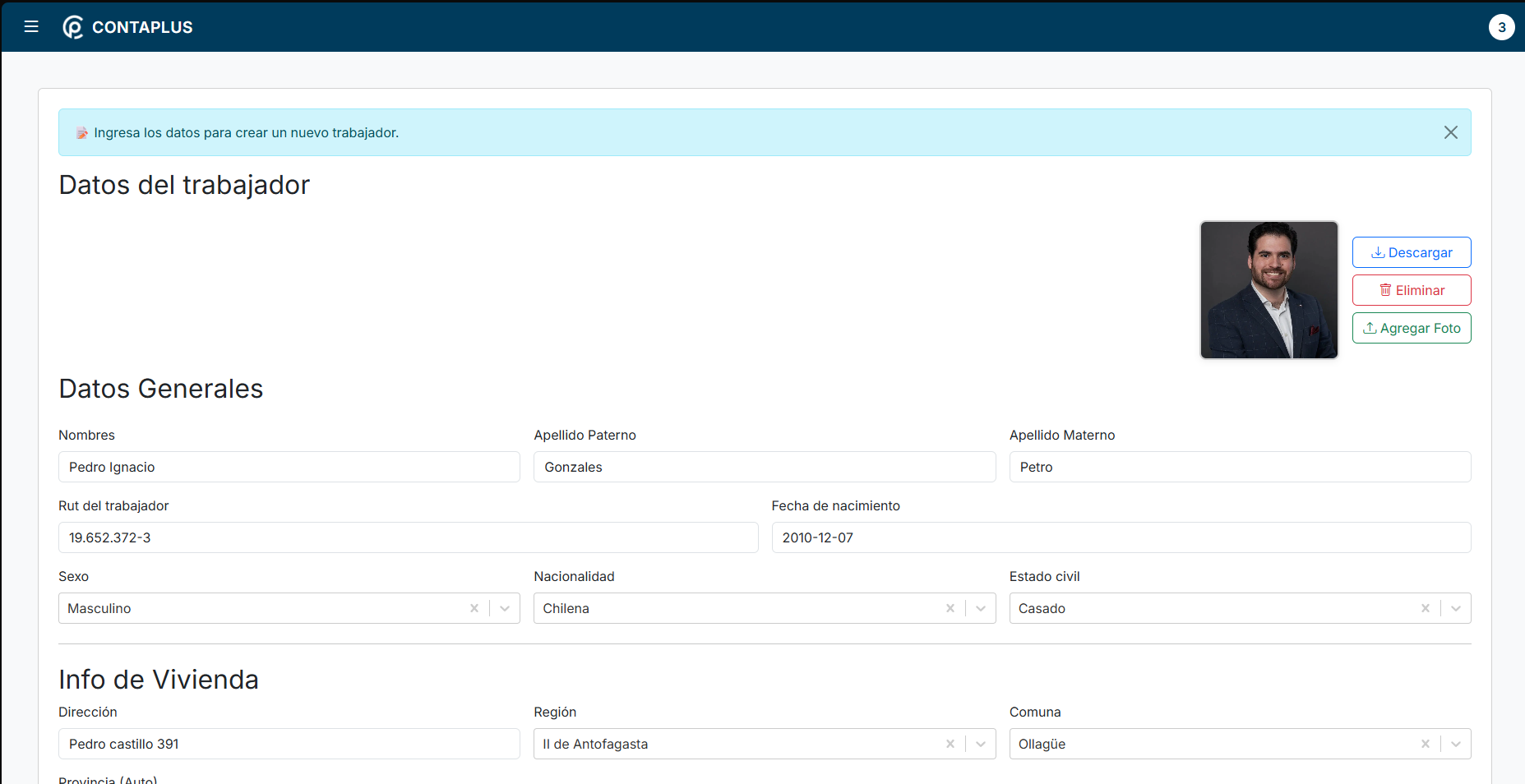
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.11 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.12 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°21:** Ingresando datos de contrato

**Casos de prueba:**

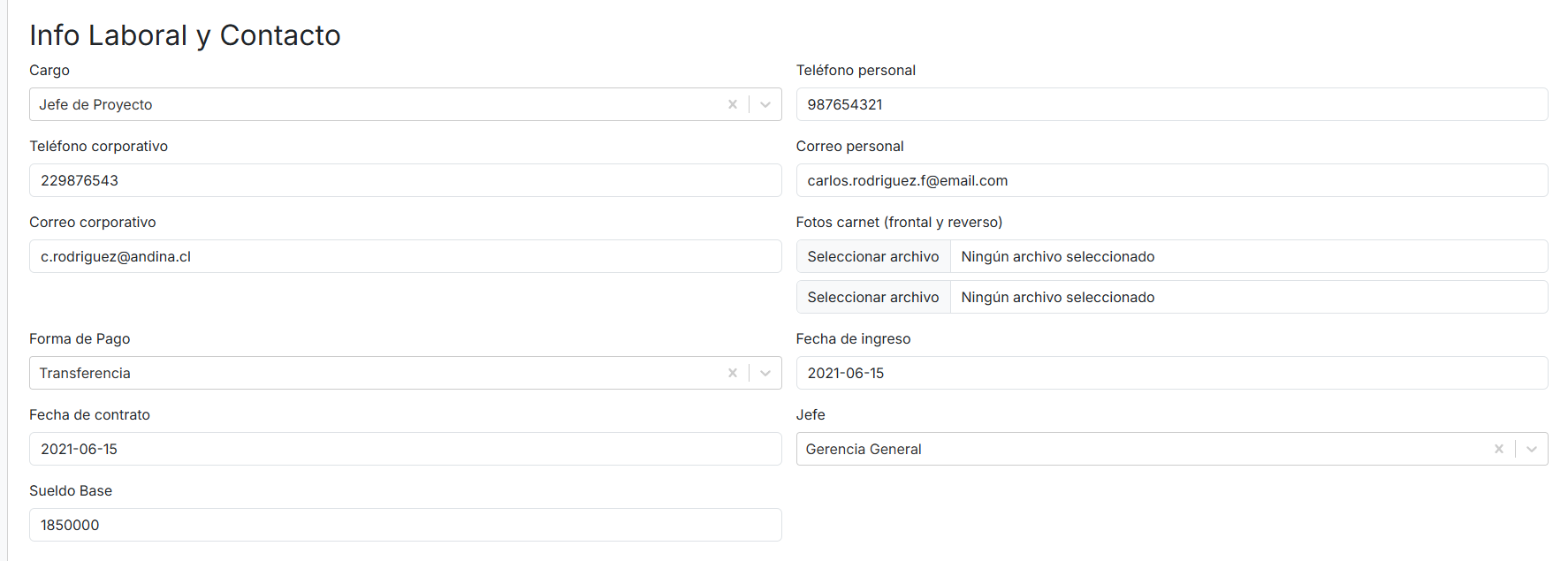
1. Ingresar al módulo de Ver Trabajadores
2. Rellenar los datos del contrato
3. Subir pdf del contrato
4. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de Ver trabajador se renderiza correctamente
2. Se rellenan correctamente los datos del contrato del trabajador
3. El pdf del contrato queda subido
4. Se guardan los nuevo datos del trabajador

**Resultados obtenidos**

### *Figura* 11.13 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°22:** Ingresando datos previsionales del trabajador

**Casos de Prueba:**

1. Ingresar al módulo de Crear Trabajadores
2. Seleccionar el afp y el institutos de salud del trabajador
3. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de crear trabajadores se renderiza correctamente
2. Se despliega una lista con las afps y otra para Institutos de salud y se seleccionan correctamente
3. Se guardan los cambios

**Resultados Obtenidos**

### *Figura* 11.14 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°23:** subiendo documentos personales del trabajador

**Casos de Prueba:**

1. Ingresar al módulo de Crear Trabajadores
2. Sube los archivo en las secciones pertinente como Contrato de trabajador
3. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de crear trabajadores se renderiza correctamente
2. Se sube correctamente los archivos y se muestra en la lista de documentos subidos
3. Se guardan los archivos

**Resultados Obtenidos**

### *Figura* 11.15 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°24:** Editando perfil completo del trabajador

**Casos de Prueba:**

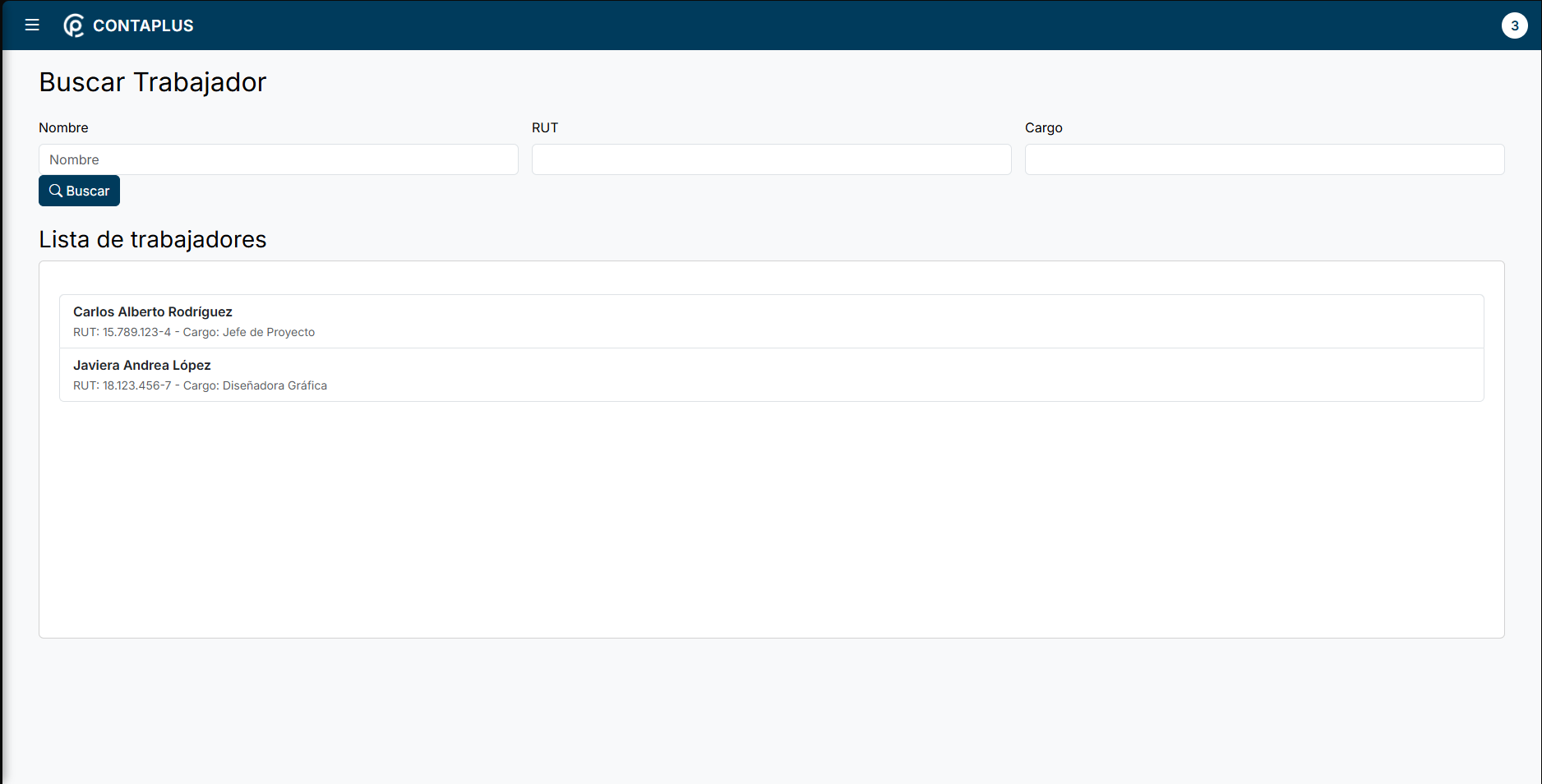
1. Ingresar al módulo de Ver trabajador
2. Pulsar el botón de Buscar
3. Seleccionar a un trabajador
4. Editar los datos del trabajador
5. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de Ver trabajador se renderiza correctamente
2. Se renderiza una lista con los trabajadores creados
3. se redirecciona a el módulo del trabajador seleccionado
4. Se editan correctamente los datos del trabajador
5. Se guardan los cambios

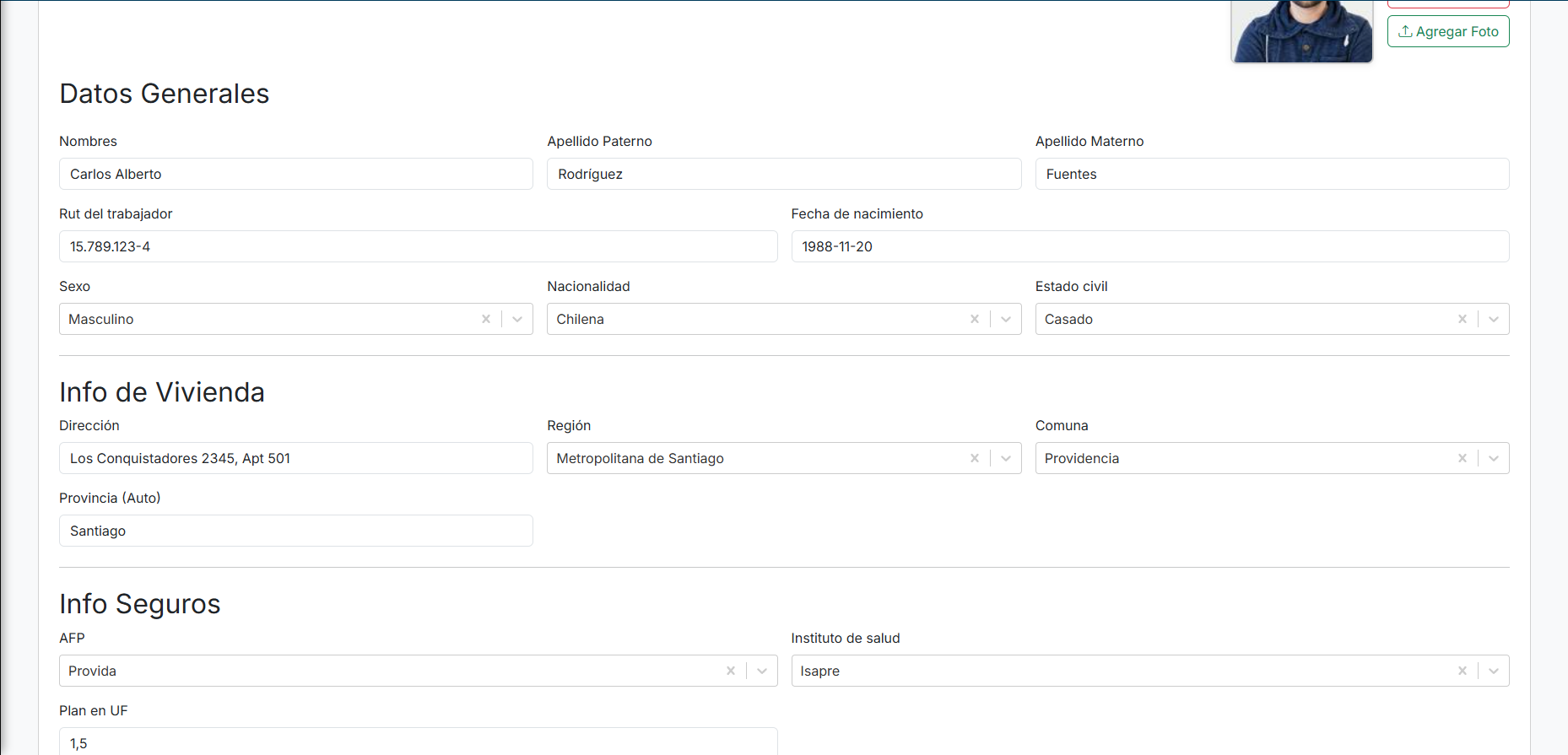
**Resultados Obtenidos:**

### *Figura* 11.13 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.14 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°25:** buscando trabajador por RUT

**Casos de Prueba:**

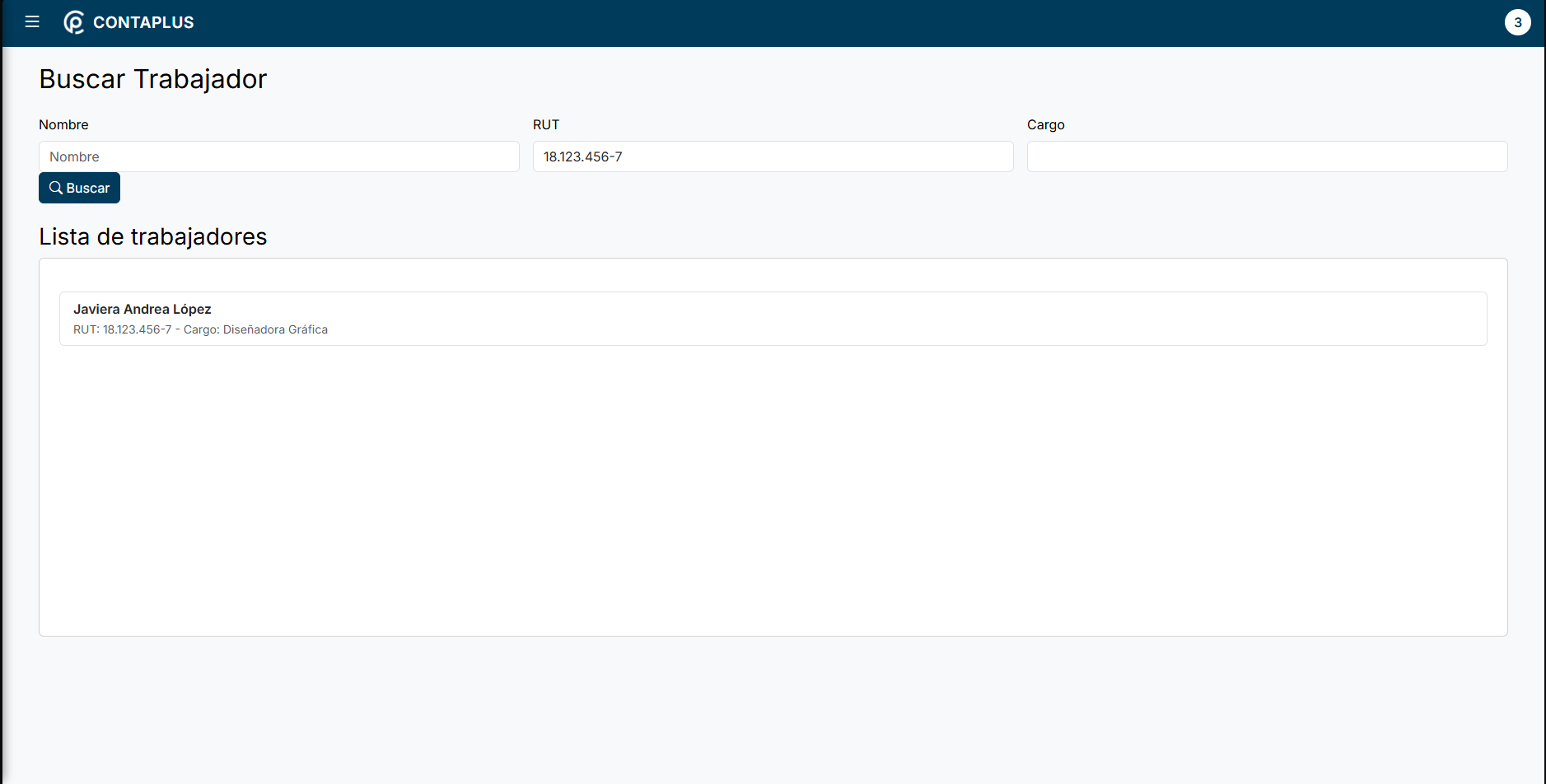
1. Ingresar al módulo de Ver trabajador
2. Ingresar el Rut del trabajador
3. Pulsar el botón de Buscar

**Resultados Esperados:**

1. El módulo de Ver trabajador se renderiza correctamente
2. Se ingresan correctamente el Rut del trabajador
3. Se renderiza la lista con el trabajador buscado

**Resultados Obtenidos:**

### *Figura* 11.15 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°26:** buscando trabajador por cargo

**Casos de Prueba:**

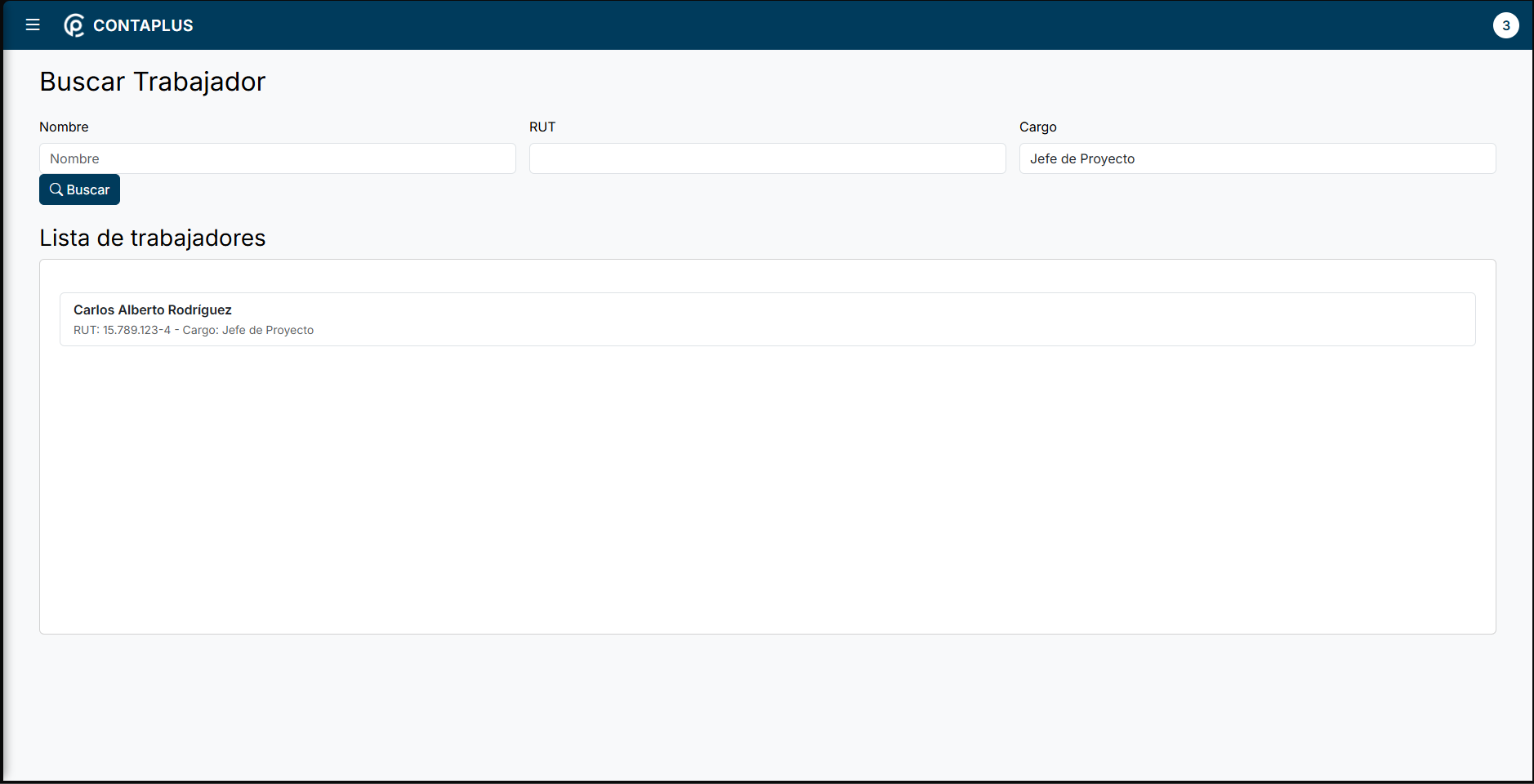
1. Ingresar al módulo de Ver trabajador
2. Ingresar el Cargo del trabajador
3. Pulsar el botón de Buscar

**Resultados Esperados:**

1. El módulo de Ver trabajador se renderiza correctamente
2. Se ingresan correctamente el Cargo
3. Se renderiza la lista con los trabajadores con el cargo ingresado

**Resultados obtenidos:**

### *Figura* 11.16 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°27:** buscando trabajador por nombre

**Casos de Prueba:**

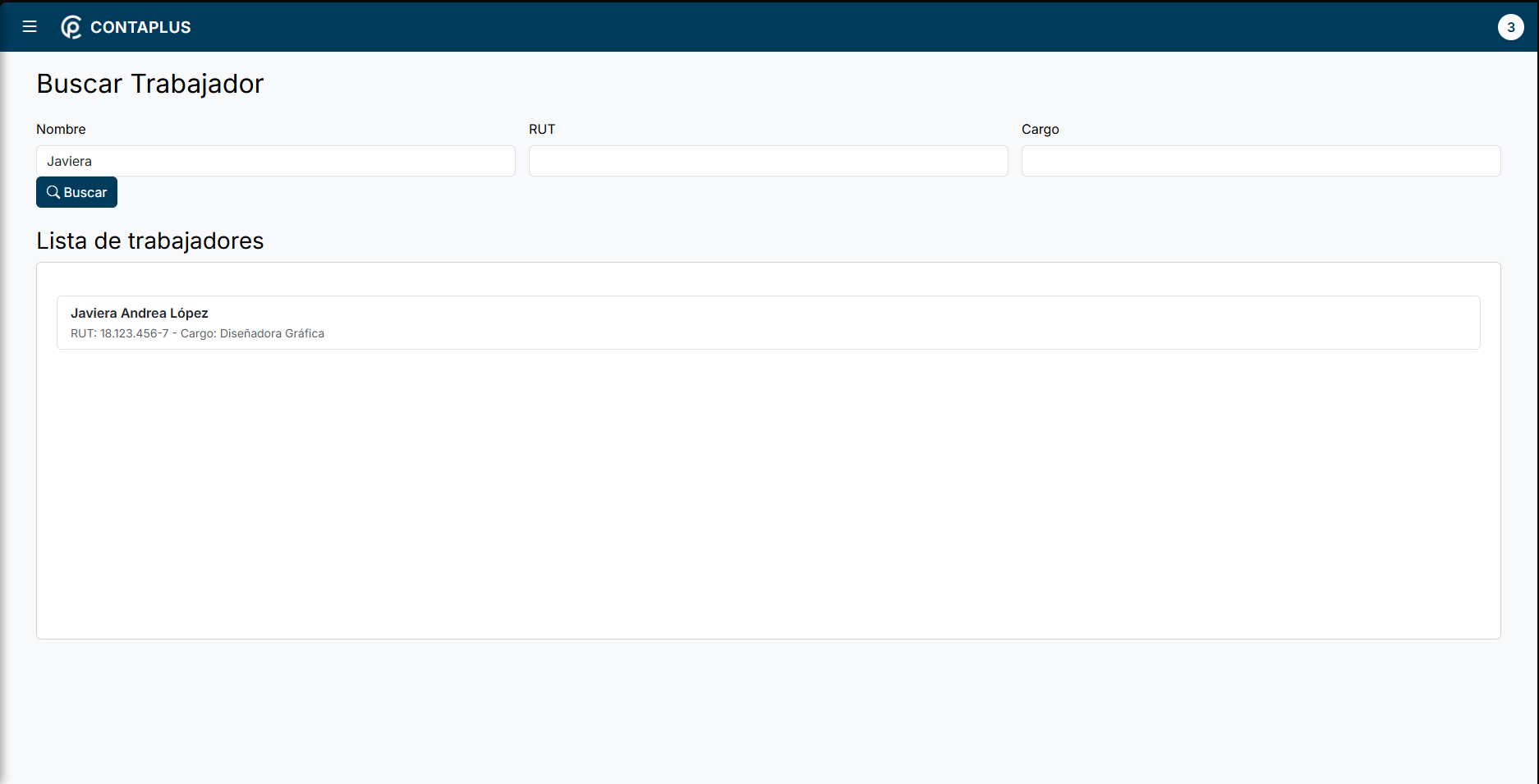
1. Ingresar al módulo de Ver trabajador
2. Ingresar el Nombre del trabajador
3. Pulsar el botón de Buscar

**Resultados Esperados:**

1. El módulo de Ver trabajador se renderiza correctamente
2. Se ingresan correctamente el nombre del trabajador
3. Se renderiza la lista con el trabajador buscado

**Resultados Obtenidos:**

### *Figura* 11.17 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°28:** visualizando perfil del trabajador

**Casos de Prueba:**

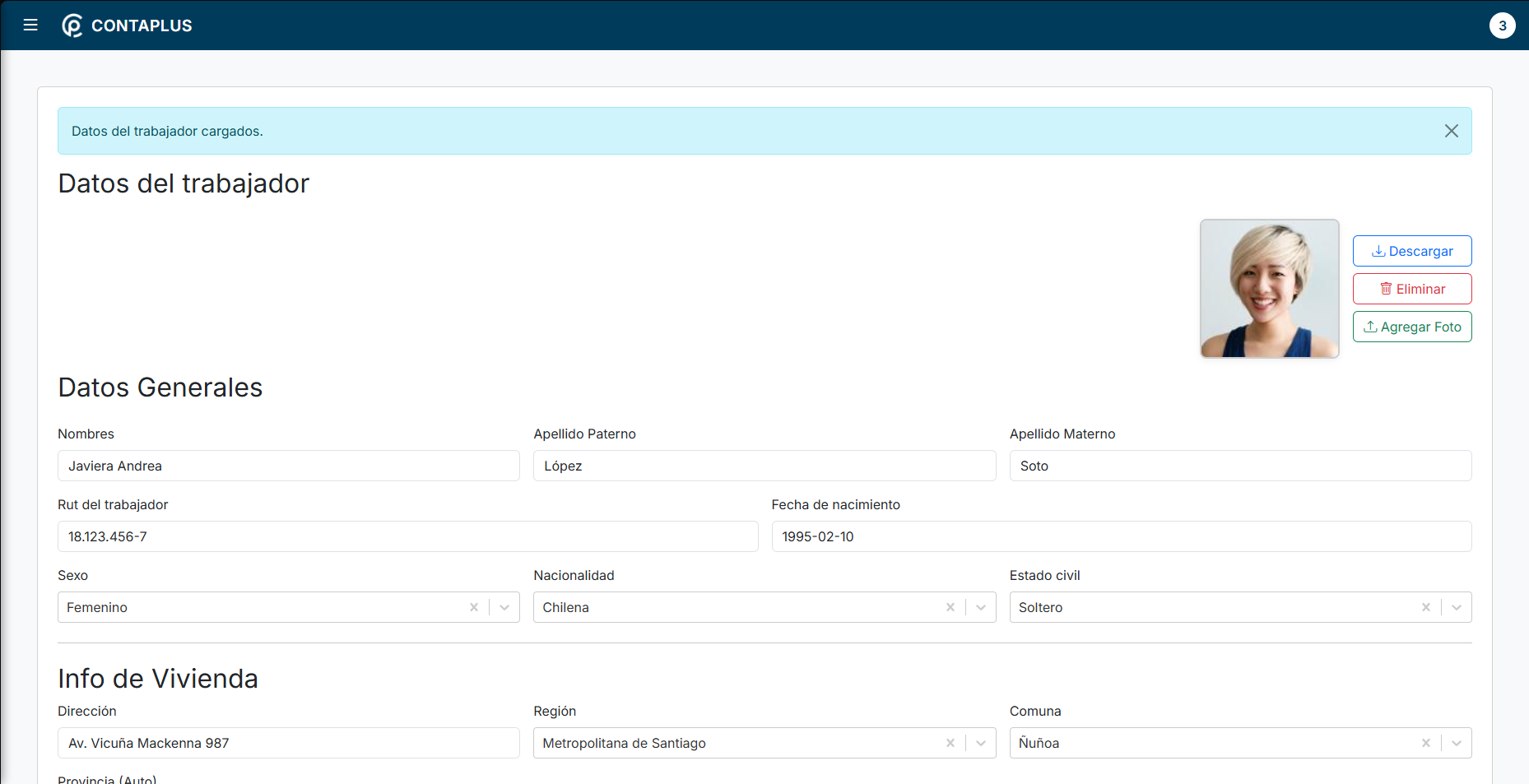
1. Ingresar al módulo de Ver trabajador
2. Pulsar el botón de Buscar
3. Seleccionar a un trabajador

**Resultados Esperados:**

1. El módulo de Ver trabajador se rendiza correctamente
2. Se renderiza la lista con los trabajadores ingresado
3. Se redirecciona al módulo de los datos del trabajador seleccionado y se renderiza con los datos del trabajador

**Resultados Obtenidos:**

### *Figura* 11.18 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°29:** Editando datos previsionales

**Casos de Prueba:**

1. Ingresar al módulo de Ver trabajador
2. Pulsar el botón de Buscar
3. Seleccionar a un trabajador
4. Seleccionar el afp y el institutos de salud del trabajador a cambiar
5. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de Ver trabajador se rendiza correctamente
2. Se renderiza la lista con los trabajadores ingresado
3. Se redirecciona al módulo de los datos del trabajador seleccionado y se renderiza con los datos del trabajador
4. Se renderiza una lista con los nombres de las afps y de los institutos de Salud y se seleccionan correctamente
5. Se guardan los cambios

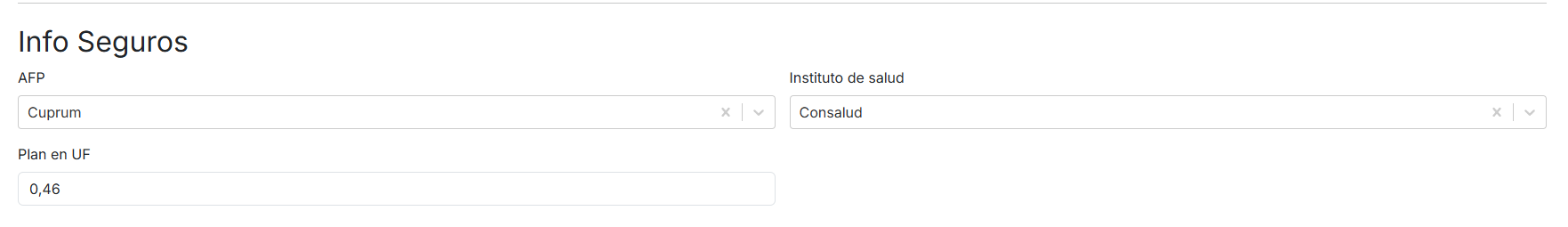
**Resultados Obtenidos:**

### *Figura* 11.19 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.20 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°30:** creando cargo

**Casos de Prueba:**

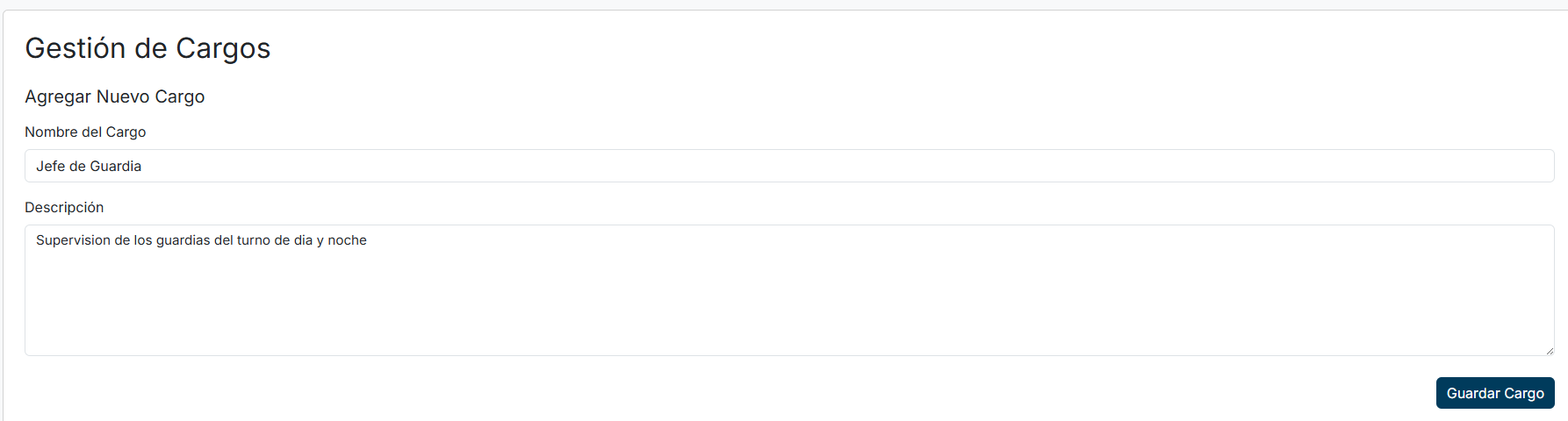
1. Ingresar al módulo REC CAP CARGO
2. Rellenar el campos de nombre de cargo
3. Rellenar el campo de descripción
4. Pulsar el botón de Guardar Cargo

**Resultados Esperados:**

1. El módulo de REC CAP CARGO se renderiza correctamente
2. Se rellena el campo de nombre del cargo de forma correcta
3. Se rellena el campo de descripción de forma correcta
4. Se guardan el cargo

**Resultados Obtenidos**:

### *Figura* 11.21 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°31:** Buscando cargo por nombre

**Casos de Prueba:**

1. Ingresar al módulo REC CAP CARGO
2. rellenar el campo de buscar por nombre de cargo

**Resultados Esperados:**

1. El módulo de REC CAP CARGO se renderiza correctamente
2. Se rellena el campo de Buscar por nombre de cargo y se renderiza una lista con el cargo buscado

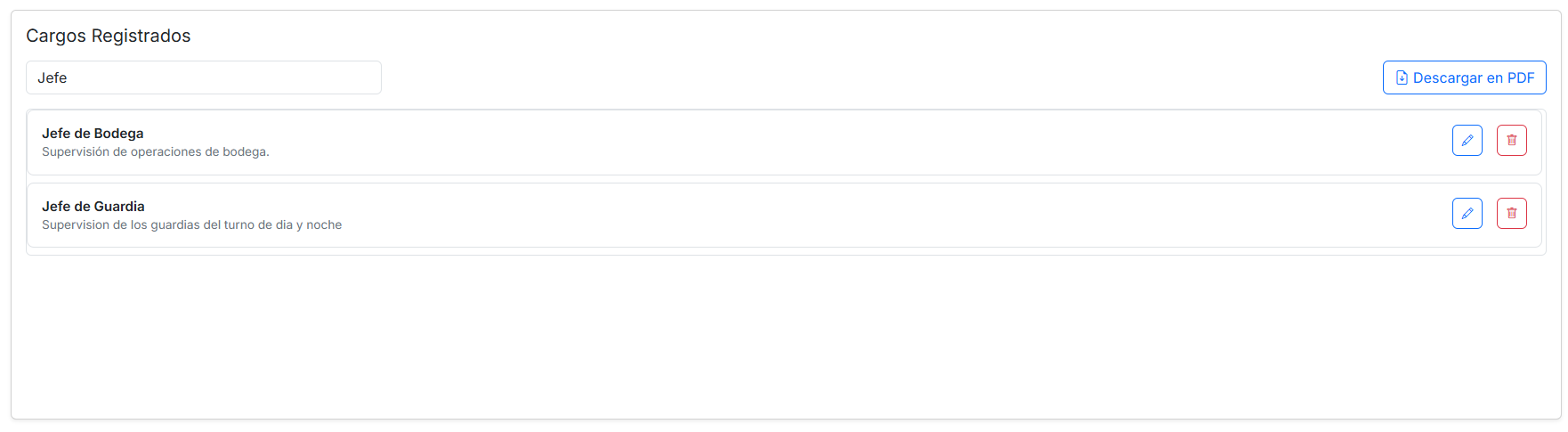
**Resultados Obtenidos:**

### *Figura* 11.22 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.23 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°32:** descargando descripción de cargo en formato .PDF

**Casos de Prueba:**

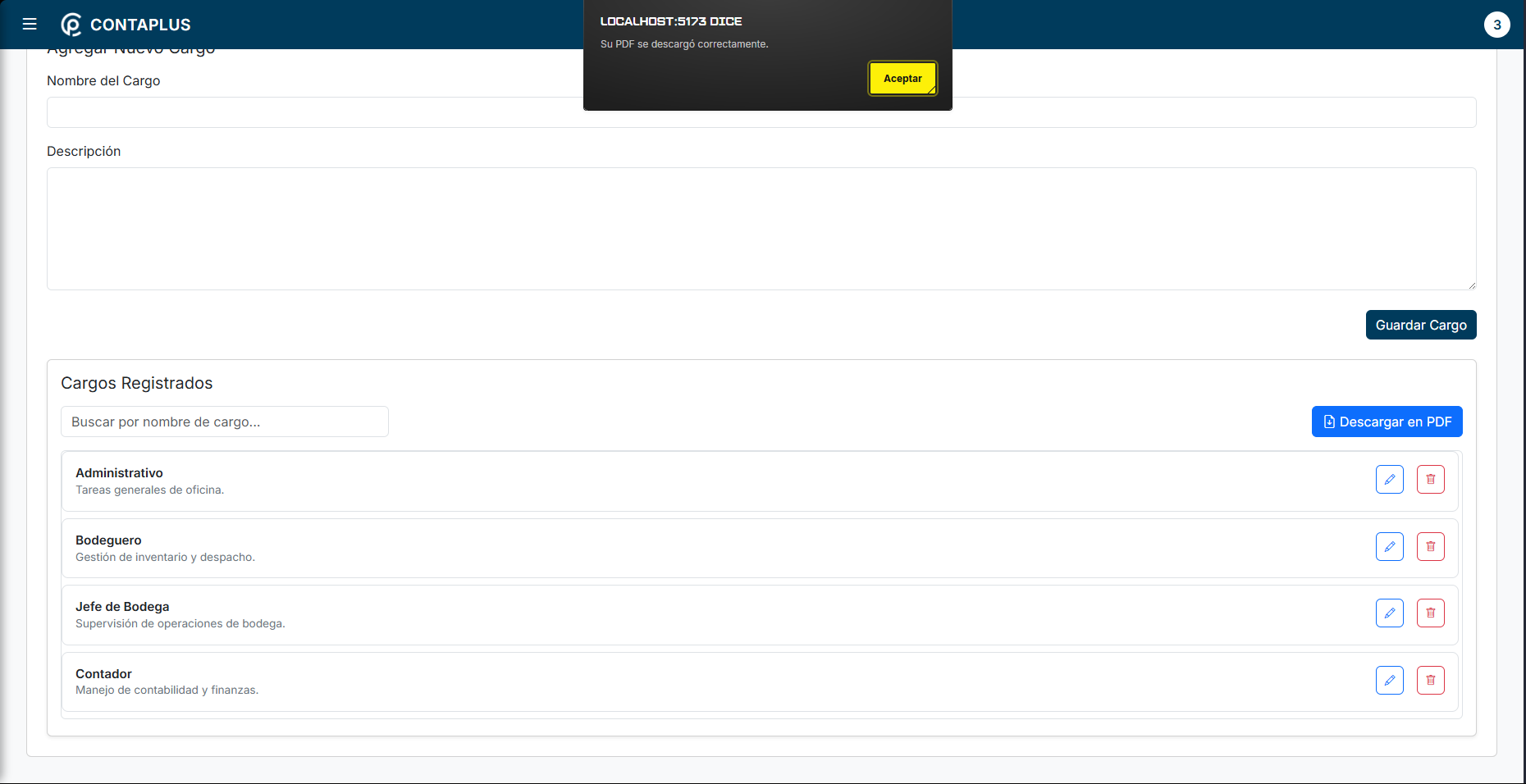
1. Ingresar al módulo REC CAP CARGO
2. Pulsar el botón de Descargar en PDF

**Resultados Esperados:**

1. El módulo de REC CAP CARGO se renderiza correctamente
2. Se descarga el PDF

**Resultados obtenidos**

### *Figura* 11.24 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°34:** Visualizando plantilla de contrato

**Casos de Prueba:**

1. Ingresar al módulo de Crear Contrato
2. Seleccionar una plantilla
3. Bajar hasta Preview del contrato

**Resultados Esperados:**

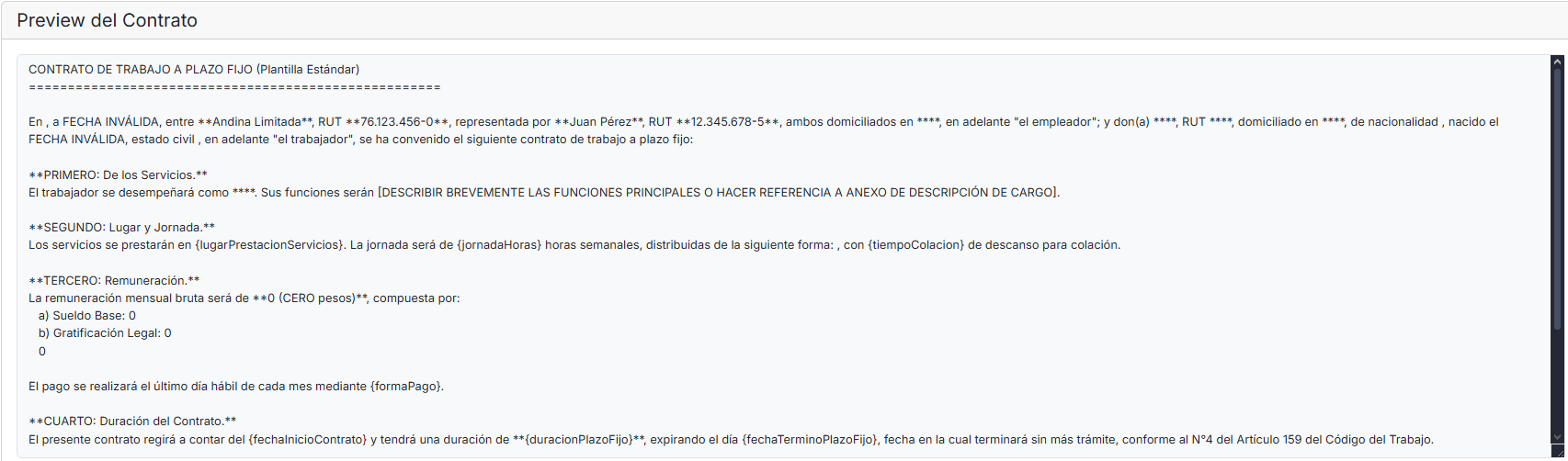
1. El módulo de Crear contrato se renderiza correctamente
2. Se renderiza la lista con las plantillas disponibles
3. En preview del contrato se podrá ver la plantilla seleccionada

Resultados Obtenidos:

### *Figura* 11.25 *“Pruebas del sistema”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.26 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°38:** Agregando anexo de contrato

**Casos de Prueba:**

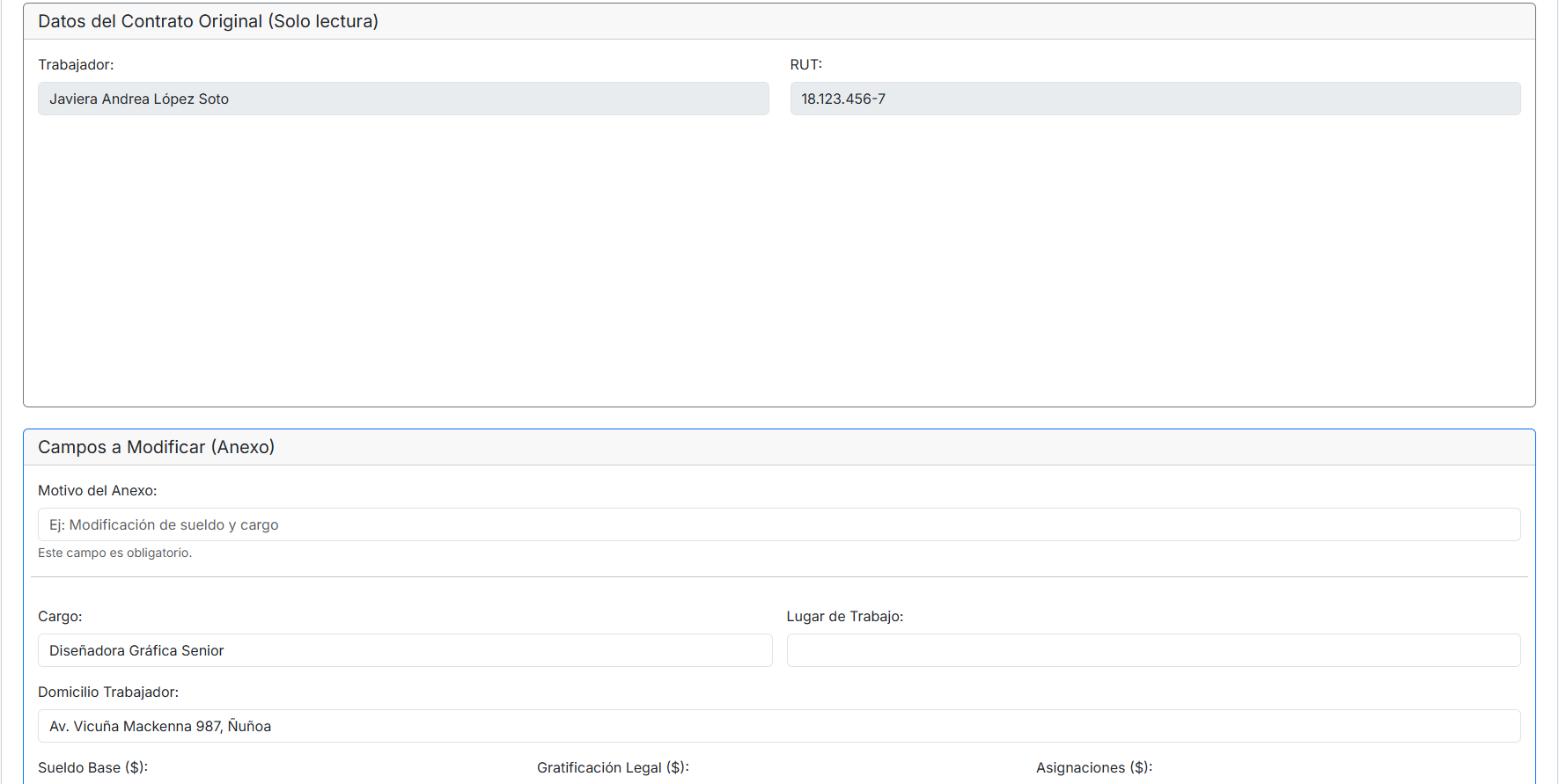
1. Ingresar al módulo de Listar Contratos
2. Pulsar el botón de Buscar
3. Pulsar el botón de Crear Anexo en el trabajador seleccionado
4. Rellenar los campos a modificar
5. Pulsar el botón de Guardar Anexo

**Resultados Esperados:**

1. El módulo de Listar Contratos se renderiza correctamente
2. Se renderiza la lista con los contratos generados
3. se redirecciona al módulo de crear anexo del contrato seleccionado
4. Se rellenan los campos de forma correcta
5. Se guarda el anexo generado

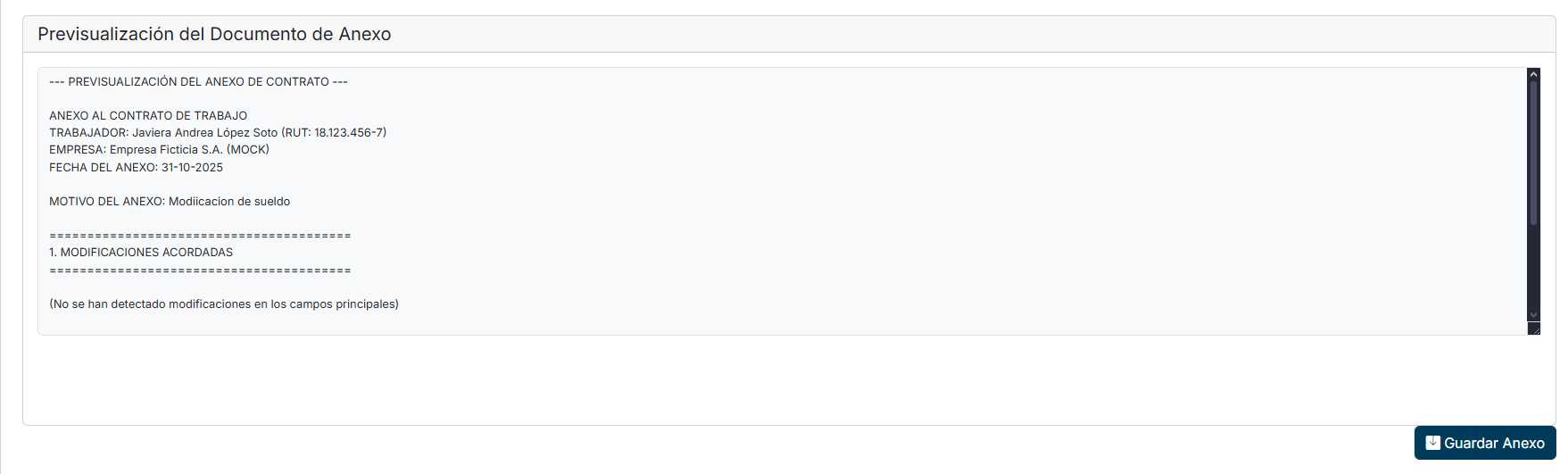
Resultados obtenidos:

### *Figura* 11.27 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.28 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°43:** Creando O.D.I

**Casos de Prueba:**

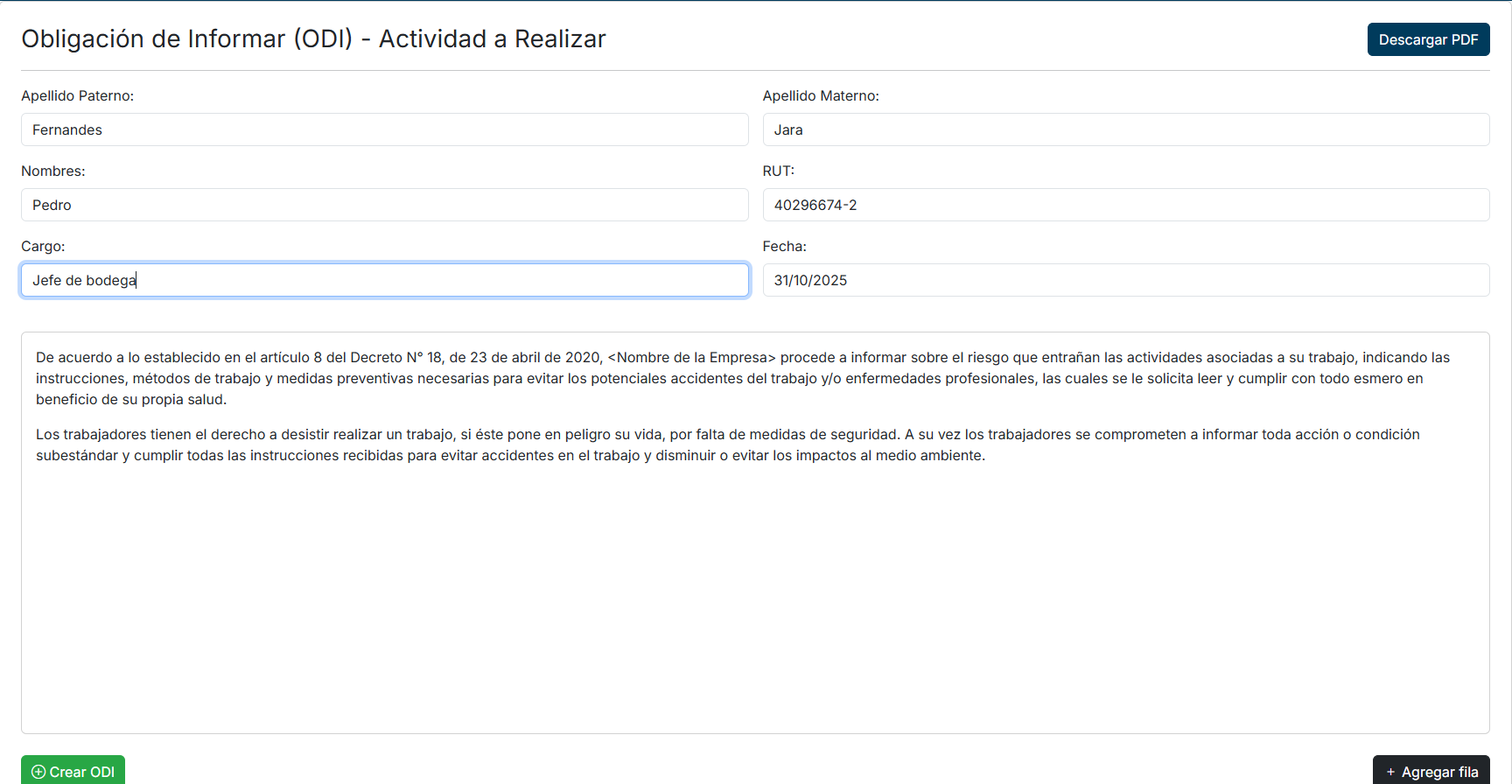
1. Accede al módulo de “REC ODI”
2. Rellena los datos del trabajador
3. Clic en “Crear ODI”
4. Rellena los datos solicitados y presiona en guardar
5. Clic en Descargar PDF

**Resultados Esperados:**

1. Muestra la vista del módulo
2. Renderiza la previsualización
3. muestra la vista de “Crear ODI”
4. Muestra los datos en la vista
5. se descarga el pdf, además de guardarlo en el sistema

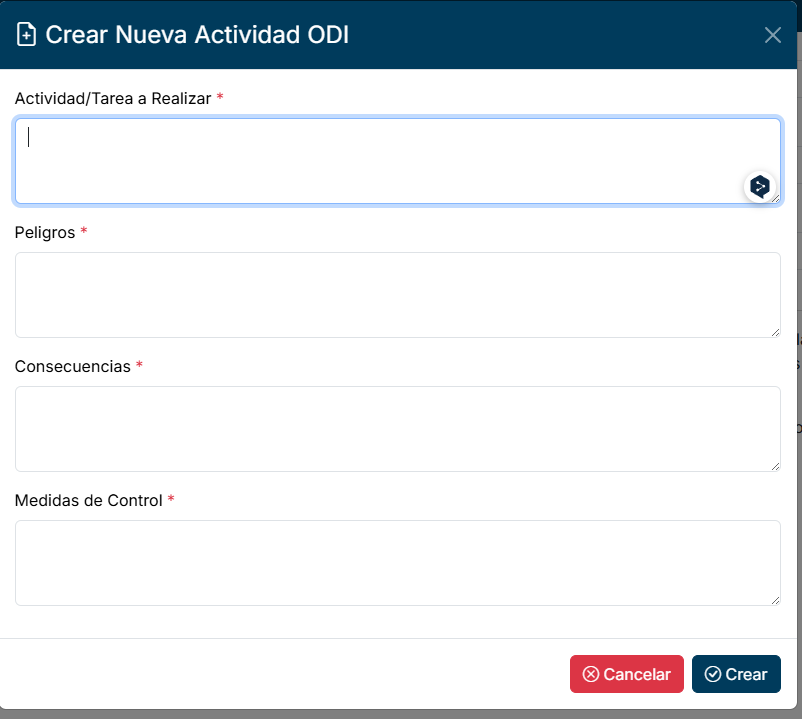
**Resultados Obtenidos**

### *Figura* 11.29 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

### *Figura* 11.30 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°44:** Subiendo pdf de O.D.I a perfil del trabajador

**Casos de Prueba:**

1. Ingresar al módulo de Ver trabajador
2. Pulsar el botón de Buscar
3. Seleccionar a un trabajador
4. Sube los archivo en la sección de ODI
5. Pulsar el botón de Guardar

**Resultados Esperados:**

1. El módulo de Ver trabajador se rendiza correctamente
2. Se renderiza la lista con los trabajadores ingresado
3. Se redirecciona al módulo de los datos del trabajador seleccionado y se renderiza con los datos del trabajador
4. Se sube correctamente el archivo ODI y se actualiza la lista de los archivos ODI subidas
5. Se guardan los cambios

Resultados obtenidos

### *Figura* 11.31 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°45:** Descargando pdf de O.D.I del perfil del trabajador

**Casos de Prueba:**

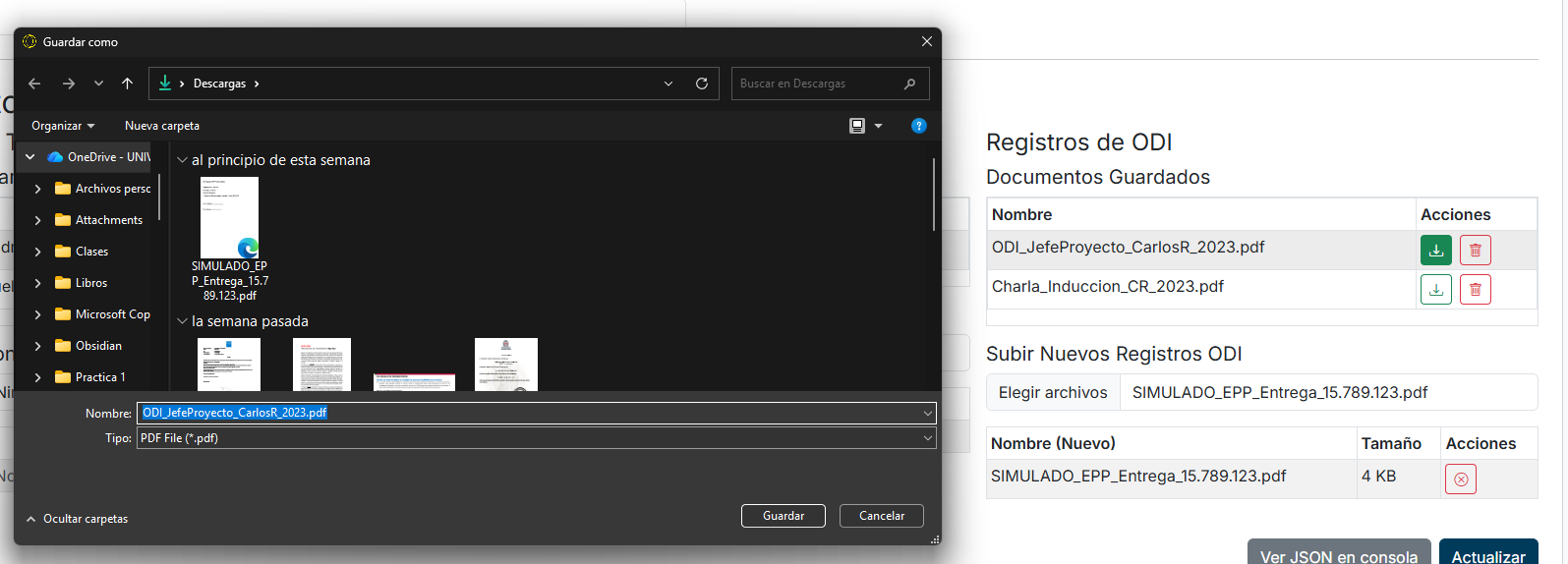
1. El módulo de Ver trabajador se rendiza correctamente
2. Se renderiza la lista con los trabajadores ingresado
3. Se redirecciona al módulo de los datos del trabajador seleccionado y se renderiza con los datos del trabajador
4. Pulsar el botón de descarga del Archivo ODI seleccionado

**Resultados Esperados:**

1. El módulo de Ver trabajador se rendiza correctamente
2. Se renderiza la lista con los trabajadores ingresado
3. Se redirecciona al módulo de los datos del trabajador seleccionado y se renderiza con los datos del trabajador
4. Se descarga el archivo ODI seleccionado

Resultado obtenido:

### *Figura* 11.32 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°46:** Descargando pdf de O.D.I filtrado con fecha

**Casos de Prueba:**

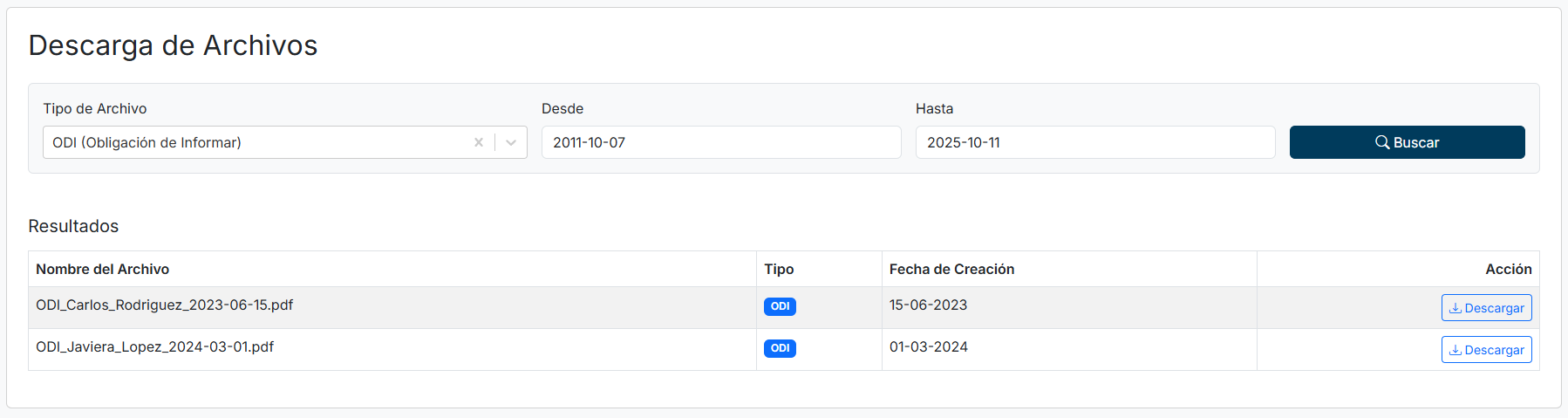
1. Ingresar al módulo de Descarga de Archivo
2. Seleccionar las fechas para filtrar
3. pulsar el botón de Buscar
4. Pulsar el botón de descargar del ODI seleccionado

**Resultados Esperados:**

1. El módulo de Descarga de Archivo se renderiza correctamente
2. Se permite seleccionar las fechas de filtro de inicio y de final
3. Se renderiza la lista con los archivos generados en el rango de esas fechas
4. Se descarga el archivo ODI seleccionado

Resultados obtenidos

### *Figura* 11.33 *“Pruebas del sistema”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°47:** Descargando pdf de contrato filtrado con fecha

**Casos de Prueba:**

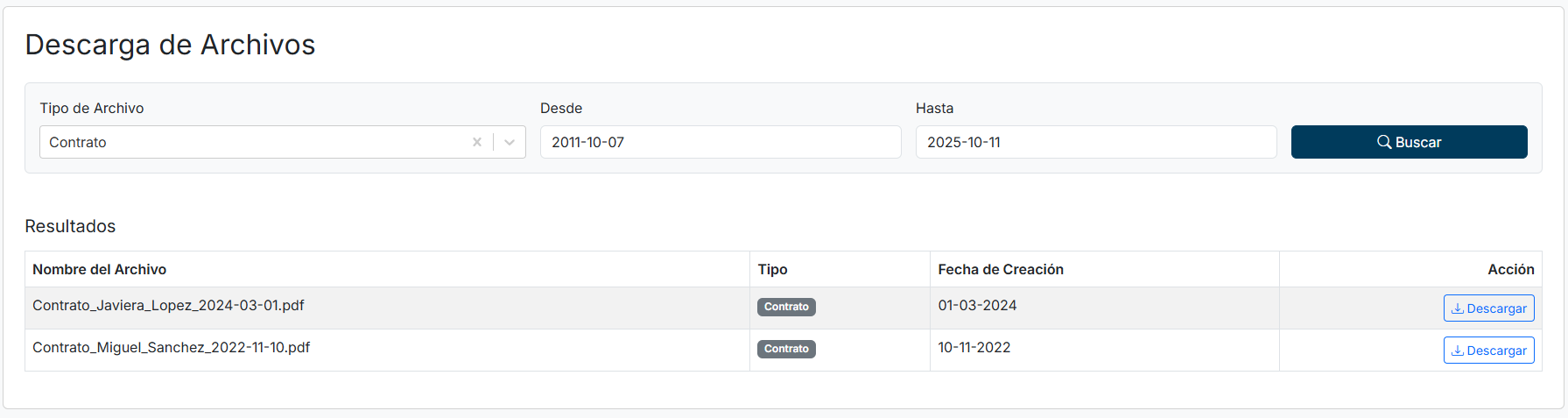
1. Ingresar al módulo de Descarga de Archivo
2. Seleccionar las fechas para filtrar
3. pulsar el botón de Buscar
4. Pulsar el botón de descargar del Contrato seleccionado

**Resultados Esperados:**

1. El módulo de Descarga de Archivo se renderiza correctamente
2. Se permite seleccionar las fechas de filtro de inicio y de final
3. Se renderiza la lista con los archivos generados en el rango de esas fechas
4. Se descarga el archivo de contrato seleccionado

**Resultados obtenidos:**

### *Figura* 11.34 *“Pruebas del sistema”*

****

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

**Caso de Uso N°48:** Descargando pdf de E.P.P filtrado por fecha

**Casos de Prueba:**

1. Ingresar al módulo de Descarga de Archivo
2. Seleccionar las fechas para filtrar
3. pulsar el botón de Buscar
4. Pulsar el botón de descargar del EPP seleccionado

**Resultados Esperados:**

1. El módulo de Descarga de Archivo se renderiza correctamente
2. Se permite seleccionar las fechas de filtro de inicio y de final
3. Se renderiza la lista con los archivos generados en el rango de esas fechas
4. Se descarga el archivo EPP seleccionado

Resultados obtenidos

### *Figura* 11.35 *“Pruebas del sistema”*

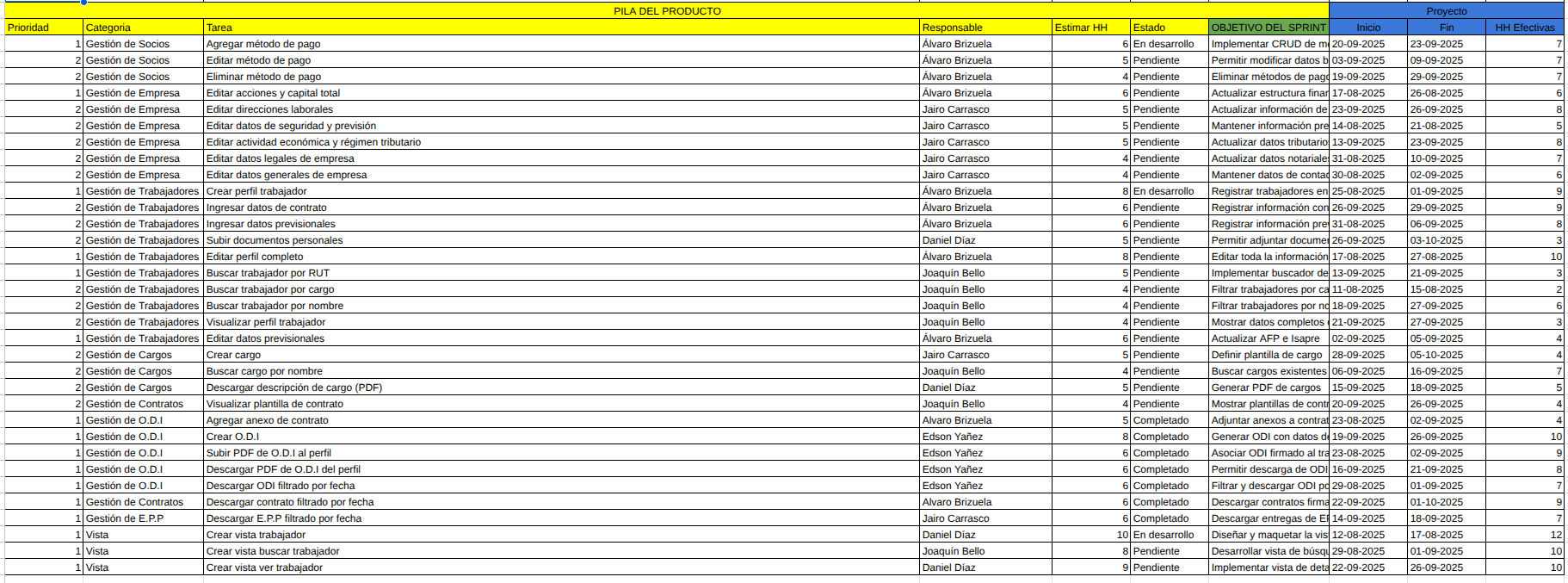
Fuente: Elaborado por el estudiante de acuerdo con el proyecto

# **12. TABLA BACKLOG CON TIEMPO REAL, TABLA DE ESFUERZO, GRÁFICO Y GRÁFICO BURN-DOWN**

A continuación, en la Tabla 12.1 se muestra el backlog del producto, que incluye una lista completa de las tareas planificadas para el desarrollo del sistema. Esta tabla nos permite ver el detalle de cada tarea, incluyendo su prioridad, categoría, responsable, esfuerzo estimado en horas, estado de avance y la fecha de planificación correspondiente.

Como se puede apreciar, el backlog incluye tareas organizadas en varias categorías como Capa vista, Capa controlador, Desarrollo de base de datos, Desarrollo de conexiones y Documentación. Cada una de estas tareas está vinculada a un objetivo específico del Sprint, como "Generar la plataforma de Datos de empresa" o "Generar Documentación".

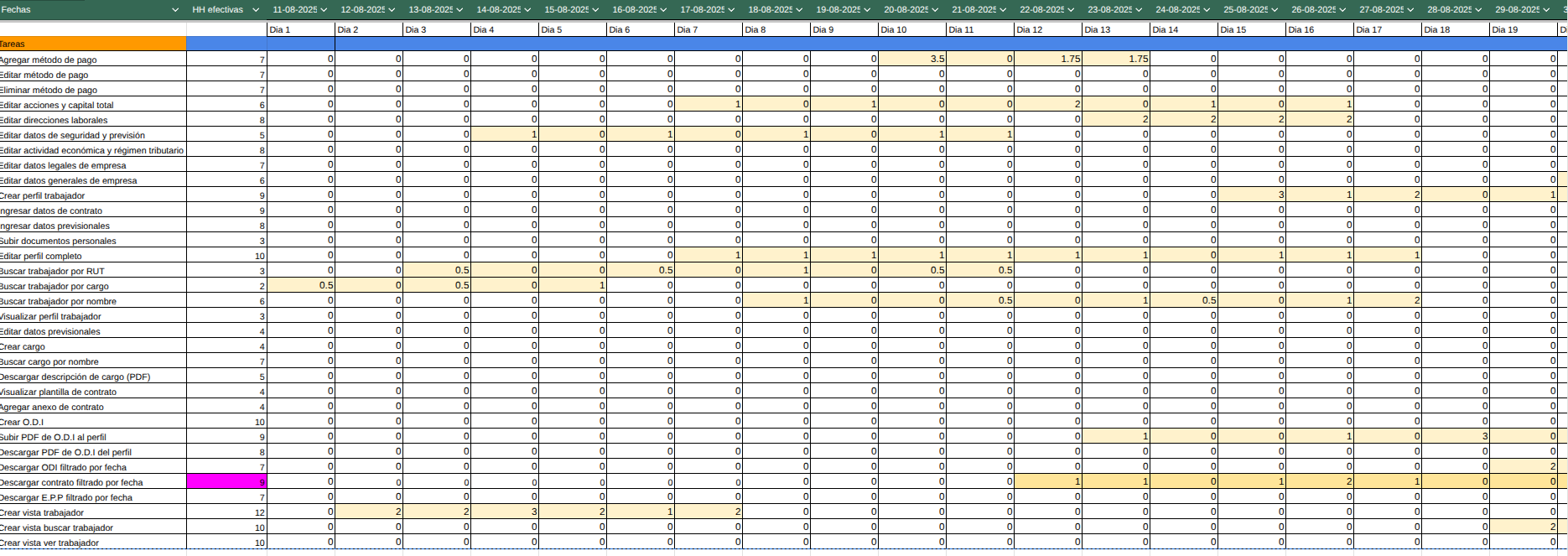
##### *Tabla* 12.1 *– “Tabla backlog con tiempo real”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

A continuación, en la Tabla 12.2 se presenta la tabla de esfuerzo, la cual permite visualizar el esfuerzo horario distribuido por tarea a lo largo del período de ejecución. En ella se muestra la cantidad de horas dedicadas diariamente por cada integrante del equipo, asociadas a las tareas del backlog y agrupadas según su categoría y objetivo del sprint.

##### *Tabla* 12.2 *– “Tabla de esfuerzo”*

Fuente: Elaborado por el estudiante de acuerdo con el proyecto

A continuación, en la Figura 12.3 se presenta la gráfica burn-down, la cual permite visualizar el comportamiento del backlog a lo largo del tiempo, en términos del número de tareas y horas pendientes. Esta gráfica refleja el seguimiento del avance del equipo durante la ejecución del Sprint, comparando el esfuerzo estimado con el esfuerzo realmente consumido a medida que se completaron las tareas.

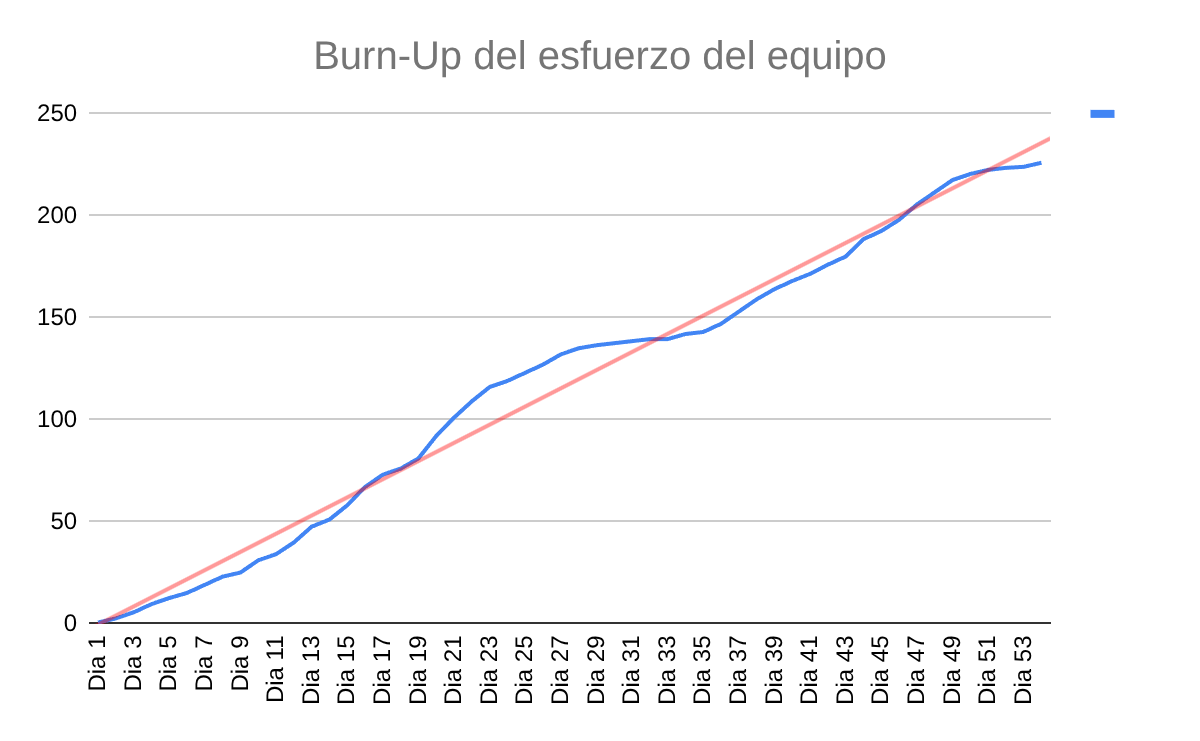
### *Figura* 12.3 *– “Diagrama de burn-down”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto

A continuación, en la Figura 12.4 se presenta la gráfica burn-up, la cual permite visualizar el progreso del backlog a lo largo del tiempo, en términos del número de tareas y horas completadas. Esta gráfica refleja el seguimiento del trabajo acumulado por el equipo durante la ejecución del Sprint, comparando el esfuerzo total planificado (alcance) con el esfuerzo realmente consumido a medida que se completaron las tareas.

### *Figura* 12.4 *– “Diagrama de burn-down”*



Fuente: Elaborado por el estudiante de acuerdo con el proyecto