



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**  
**FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

# **"Especificación Formal y Validación de un Sistema de cobro de pasajes en el transporte público utilizando Smart Card con la tecnología NFC"**

**Autores:**

- **Chura Puma, Mario Franco**
- **Llaique Chullunquia, Angie Carolina**
- **Tacca Apaza, Nohelia Estefhania**

# Introducción

- El transporte público en Arequipa enfrenta problemas por la eliminación de cobradores durante la pandemia
- Se propone un sistema de cobro automatizado con tarjetas NFC para agilizar el pago y reducir la evasión
- Se realizará la especificación formal de los requerimientos en VDM++ y se validará el modelo propuesto
- El objetivo es mejorar la eficiencia y seguridad del transporte público en la ciudad





# Marco teórico



# Tecnología NFC

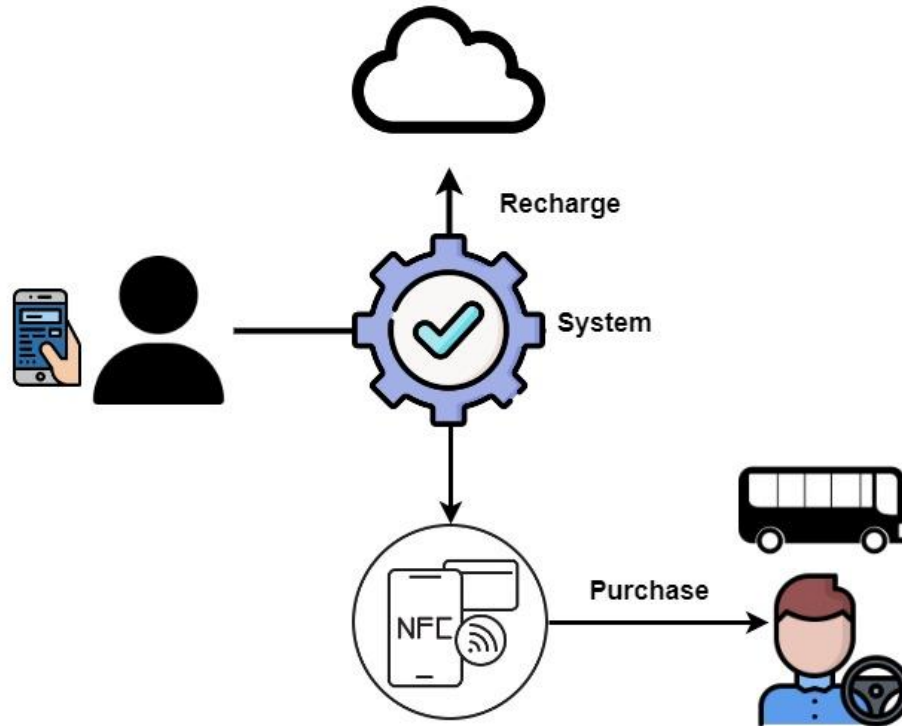


Establecimiento de la comunicación	Menor a 0,1 seg
Velocidad de transmisión	424 kbps 848 kbps
Alcance	Menor a 10 cm
Consumo de batería	Bajo
Seguridad	Alta
Experiencia en usuarios	Simplemente con un toque



# Propuesta

# Arquitectura



# Requerimientos

**R1:** Cada código de usuario debe ser único.

**R2:** Cada código de Smart Card debe ser único.

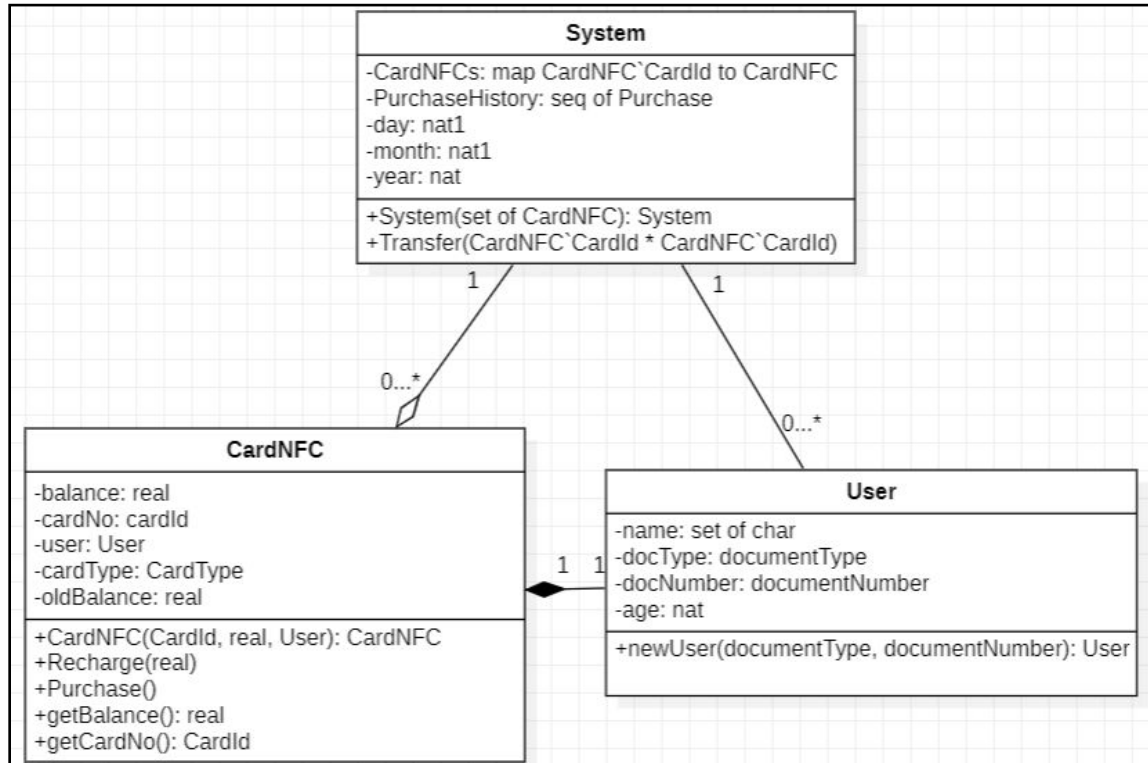
**R3:** No se permitirán pagos si es que no se cuenta con el saldo suficiente.

**R4:** El saldo resultante después de una recarga, debe de ser igual al saldo anterior más el monto de la recarga.

**R5:** Las tarjetas con las cuales se realizan las transacciones deben de estar registradas en el sistema.

**R6:** El id de la tarjeta que realiza el pago y el id de la tarjeta que recibe el pago deben de ser diferentes.

# Diagrama de clases







# **Especificaciones Formales**



# Clases en VDM++

## Clase User

```
1: class User
2: types
3:   public documentNumber = token;
4:   public documentType = <DNI> | <Passport>;
5:
6: instance variables
7:   private name: seq of char;
8:   private docType: documentType;
9:   private docNumber: documentNumber;
10:  private age: nat;
11:
12: operations
13:  public User: documentType * documentNumber ==> User
14:    User(userDocType, newId) ==
15:    (docType:= userDocType;
16:     docNumber:= newId;
17:     return self
18:    )
19:
20: end User
```

# Clases en VDM++

```
1: class CardNFC
2:
3: types
4: public CardId = token;
5: public CardType = <scholar> | <general> | <personalized>;
6:
7: values
8: public ticketPrice: real = 1;
9:
10: instance variables
11: private balance: real;
12: private cardNo: CardId;
13: private user: User;
14: private cardType: CardType;
15: private oldBalance: real := 0;
16:
17: operations
18: -Constructor
19: public CardNFC : CardId * real * User ==> CardNFC
20: CardNFC (newId, startbal, newUser) ==
21:   (cardNo := newId;
22:    balance := startbal;
23:    user := newUser);
24:
25: -Recargas
26: public Recharge: real ==> () -recharge
27:   Recharge(amount) ==
28:     (oldBalance := balance;
29:      balance := balance + amount)
30:   post balance = oldBalance + amount; - R4
31:
```

## Clase Card

```
31:
32: -Pagos de pasajes
33: public Purchase: () ==> () - purchase
34: Purchase() ==
35:   balance := balance - ticketPrice
36:   pre ticketPrice <= balance; - R3
37:
38: -Ver saldo
39: public GetBalance: () ==> real
40: GetBalance() ==
41:   return balance;
42:
43: - Ver id del smart card
44: public GetCardNo: () ==> CardId
45: GetCardNo() ==
46:   return cardNo;
47:
48: end CardNFC
```

# Clases en VDM++

```
1: class System
2:
3: values
4: public amount : real = CardNFC.ticketPrice;
5:
6: types
7: public Purchase ::
8:   fromId : CardNFC*CardId
9:   told : CardNFC*CardId
10:   amount : real
11:   idPurchase : token;
12:
13: instance variables
14:
15: private CardNFCs: map CardNFC*CardId to CardNFC;
16: private PurchaseHistory: seq of Purchase := [];
17: public day : nat 1 := 1;
18: public month : nat 1 := 1;
19: public year : nat := 2000;
20:
21: inv forall p in set dom CardNFCs & CardNFCs(p).GetCardNo() = p; --
22: inv day >= 1 and day <= 31 and
23: month >= 1 and month <= 12 and
24: (if month in set {4, 6, 9, 11} then day <= 30
25: else if month = 2 then day <= 29
26: else day <= 31);
```

# Clase System

```
27:
28: operations
29: --Constructor
30: public System: set of CardNFC ==> System
31:   System(CardSet) ==
32:   CardNFCs := {p.GetCardNo() |> p | p in set CardSet} --Id |> objeto
33:
34: pre forall p,q in set CardSet & p <> q => p.GetCardNo() <> q.GetCardNo(); --R2
35:
36:
37: public Transfer: CardNFC*CardId * CardNFC*CardId ==> ()
38: Transfer(fromId, told) ==
39:   (CardNFCs(fromId).Purchase();
40:   CardNFCs(told).Recharge(amount);)
41:
42: pre {fromId, told} subset dom CardNFCs and
43:   fromId <> told and
44:   CardNFCs(fromId).GetBalance() >= amount; --R5, R6
45:
46: end System
```

# Validación

```
>> tcov reset
>> create u1 := new User(<DNI>, mk_token(71067547))
>> create u2 := new User(<Passport>, mk_token(116278131))
>> create u3 := new User(<DNI>, mk_token(02279916))
>> create c1 := new CardNFC(mk_token(71067547), 0.0, u1)
>> create c2 := new CardNFC(mk_token(116278131), 20.0, u2)
>> create c3 := new CardNFC(mk_token(02279916), 50.0, u3)
>> create s1 := new System({c1,c2, c3})
>> print(s1.Transfer(mk_token(116278131),
mk_token(71067547)))
>> print (c1.GetBalance())
>> print (c2.GetBalance())
>> print (s1.Transfer(mk_token(02279916),mk_token(71067547)))
>> print (c3.GetBalance())
>> print (c3.Recharge(15))
>> print (c3.GetBalance())
```



## Interpreter Window

```
Initializing specification ...
done
>> create u1 := new User(<DNI>, mk_token(71067547))
>> create u2 := new User(<Passport>, mk_token(116278131))
>> create u3 := new User(<DNI>, mk_token(02279916))
>> create c1 := new CardNFC(mk_token(71067547), 0.0, u1)
>> create c2 := new CardNFC(mk_token(116278131), 20.0, u2)
>> create c3 := new CardNFC(mk_token(02279916), 50.0, u3)
>> create s1 := new System({c1,c2, c3})
>> print (c1.GetBalance())
0
>> print (c2.GetBalance())
20
>> print (s1.Transfer(mk_token(02279916),mk_token(71067547)))
(no return value)
>> print (c3.GetBalance())
49
>> print (c3.Recharge(15))
(no return value)
```

```
>> tcov write vdm.tc
>> rtinfo vdm.tc
100% 3 User`User
100% 1 System`System
100% 2 System`Transfer
100% 3 CardNFC`CardNFC
100% 2 CardNFC`Purchase
100% 3 CardNFC`Recharge
100% 21 CardNFC`GetCardNo
100% 7 CardNFC`GetBalance
```

Total Coverage: 100%

# Conclusiones

Este trabajo ofrece una solución tecnológica avanzada, y contribuye a la eficiencia y calidad del servicio de transporte público. La adopción de la tecnología NFC no solo simplificará el proceso de pago, sino que también permitirá una mejor planificación y supervisión del flujo de pasajeros.

De acuerdo a la validación del sistema mediante VDM++, la cobertura es del 100%, lo cual asegura la consistencia y confiabilidad del mismo, reforzando la importancia de la aplicación de especificaciones formales en el desarrollo de software. De esta manera, se espera que la implementación del sistema propuesto contribuya de manera significativa a la mejora del sistema de transporte público, beneficiando a la comunidad en general.



**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**  
**FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

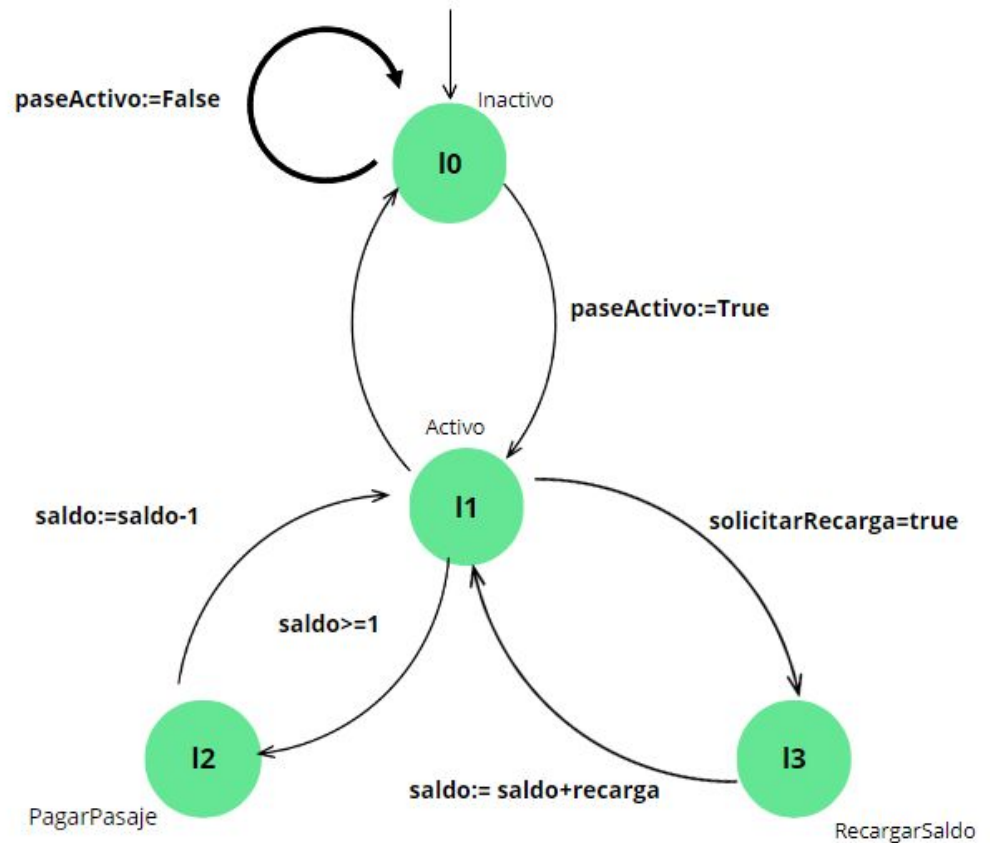
# **"Especificación Formal y Validación de un Sistema de cobro de pasajes en el transporte público utilizando Smart Card con la tecnología NFC"**

**Autores:**

- **Chura Puma, Mario Franco**
- **Llaique Chullunquia, Angie Carolina**
- **Tacca Apaza, Nohelia Estefhania**

# Diagrama de transición de estados

- Estados: Inactivo, Activo, PagarPasaje, RecargarSaldo
- Variables: saldo, recarga, solicitarRecarga, paseActivo
- Condiciones y asignaciones





```

1  MODULE main
2  VAR
3      state: {Inactivo, Activo, PagarPasaje, RecargarSaldo};
4      saldo: 0 .. 100;
5      recarga: 0 .. 10;
6      solicitarRecarga: boolean;
7      paseActivo: boolean;
8  ASSIGN
9      init(state) := Inactivo;
10     init(saldo) := 50;
11     init(recarga) := 0;
12     init(solicitarRecarga) := TRUE;
13     init(paseActivo) := FALSE;
14     next(state) := case
15         state = Inactivo & paseActivo = TRUE : Activo;
16         state = Inactivo & paseActivo = FALSE : Inactivo;
17         state = Activo & saldo >= 1 : PagarPasaje;
18         state = Activo & solicitarRecarga = TRUE : RecargarSaldo;
19         state = PagarPasaje : Activo;
20         state = RecargarSaldo : Activo;
21         TRUE : state;
22     esac;
23     next(saldo) := case
24         state = PagarPasaje & saldo > 0 : saldo - 1;
25         state = RecargarSaldo & saldo + recarga <= 1000 : saldo + recarga;
26     TRUE : saldo;
27     esac;

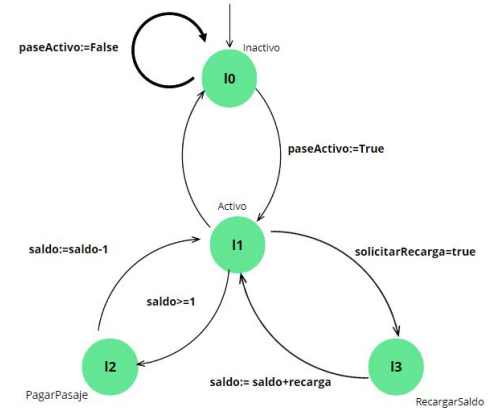
```

# 5digo

```

next(recarga) := case
    state = RecargarSaldo : 10;
    TRUE : 0;
esac;
next(solicitarRecarga) := TRUE;
next(paseActivo) := case
    state = Inactivo : TRUE;
    TRUE : FALSE;
esac;

```



# NuSMV

```
NuSMV > read_model -i systemf.smv
NuSMV > flatten_hierarchy
NuSMV > encode_variables
NuSMV > build_model
NuSMV > pick_state -i
```

```
***** AVAILABLE STATES *****
```

```
===== State =====
```

```
0) -----
```

```
state = Inactivo
```

```
saldo = 50
```

```
recarga = 0
```

```
solicitarRecarga = TRUE
```

```
paseActivo = FALSE
```

# Validación de propiedades en NuSMV

```
NuSMV > check_ltlspec -p "F (state = Activo)"  
-- specification F state = Activo is true  
NuSMV > check_ltlspec -p "F (state = Inactivo)"  
-- specification F state = Inactivo is true  
NuSMV > check_ltlspec -p "F (state = PagarPasaje)"  
-- specification F state = PagarPasaje is true  
NuSMV > check_ltlspec -p "F (state = RecargarPago)"  
-- specification F state = RecargarPago is true
```

Con la propiedad F verificamos que en algún momento pasamos por cada estado.

# Validación de propiedades en NuSMV

```
NuSMV > check_ltlspec -p "F (saldo=50)"  
-- specification F saldo = 50 is true  
NuSMV > check_ltlspec -p "F (recarga=10)"  
-- specification F recarga = 10 is true
```

También verificamos que la variable saldo en algún punto llegue a tomar el valor de 50 y que la variable recarga tome el valor de 10.

```
NuSMV > check_ltlspec -p "F (saldo=49)"  
-- specification F saldo = 49 is true
```

Por otro lado, verificamos que después de realizar un pago el saldo pase a 49.

```
NuSMV > check_ltlspec -p "G (saldo >= 0)"  
-- specification G saldo >= 0 is true  
NuSMV > check_ltlspec -p "G (recarga >= 0)"  
-- specification G recarga >= 0 is true
```

Con la propiedad G, verificamos que nuestra variable saldo, siempre sea mayor a cero.



# Conclusiones

La aplicación del model checking, respaldada por el Diagrama de Transición de Estados y la Validación de propiedades mediante NuSMV, ha sido esencial para asegurar la robustez y confiabilidad del Sistema de cobro de pasajes en el transporte público con Smart Card y tecnología NFC. Este enfoque de verificación ha permitido identificar y abordar posibles problemas antes de la implementación, garantizando la coherencia y efectividad del sistema.

# Referencias

- [1] C. E. Torres Vinuesa, "Implementación de un prototipo de sistema de pago usando la tecnología NFC para el transporte público", bachelor Thesis, Univ. Am., Quito, 2016.
- [2] Bieler, M., Skretting, A., Büdinger, P., & Grønli, T. M. (2022). Survey of Automated Fare Collection Solutions in Public Transportation. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 14248-14266.
- [3] U. Demir Alan y D. Brant, "Server-Based Intelligent Public Transportation System with NFC", *IEEE Intell. Transp. Syst. Mag.*, vol. 10, n.º 1, pp. 30-46, 2018. Accedido el 23 de septiembre de 2023. [En línea]. Disponible: <https://doi.org/10.1109/mts.2017.2776102>
- [4] Deyby, C. C. J., & Ivan, H. M. W." Especificación Formal y Validación en el Modelado de un Sistema de control en tiempo real en Instituciones de Educación Inicial utilizando Beacons", 2018.
- [5] P. Arroyo. "¿Cuánto dinero ganan los cobradores de combis en Perú durante su labor diaria?" Últimas noticias de última hora del Perú y el mundo en La República. Accedido el 9 de noviembre de 2023. [En línea]. Disponible: <https://larepublica.pe/datos-lr/2022/08/12/combis-en-peru-cuanto-dinero-gana-un-cobrador-durante-su-labor-diaria-eyat>
- [6] F. Raffo. "El transporte público era el quinto problema que más afecta a ciudadanos: ahora es el segundo". *El Comercio Perú*. Accedido el 9 de noviembre de 2023. [En línea]. Disponible: <https://elcomercio.pe/lima/el-transporte-publico-paso-a-ser-el-segundo-problema-que-mas-afecta-a-ciudadanos-lima-como-vamos-noticia/>
- [7] Huapaya Nava, M., & Soto Chávez, E. "De todos y de nadie: el caso del Transporte Público en Lima y Callao y la necesidad de un planeamiento común". *Revista De Derecho Administrativo*, (12), 323-336. Recuperado a partir de <https://revistas.pucp.edu.pe/index.php/derechoadministrativo/article/view/13513>
- [8] RPP. "Lambayeque: Transportistas anuncian incremento del pasaje en el servicio público | RPP Noticias". *RPP - Noticias del Perú y el Mundo | Radio | Podcast | RPP Noticias*. Accedido el 9 de noviembre de 2023. [En línea]. Disponible: <https://rpp.pe/peru/lambayeque/lambayeque-transportistas-anuncian-incremento-del-pasaje-en-el-servicio-publico-noticia-1361976?ref=rpp>
- [9] D. Contreras. "Movilidad sostenible: ¿Qué desafíos y problemas enfrenta Lima Metropolitana? | Instituto de la Naturaleza, Tierra y Energía (INTE-PUCP)". *Instituto de la Naturaleza, Tierra y Energía (INTE-PUCP)*. Accedido el 9 de noviembre de 2023. [En línea]. Disponible: <https://inte.pucp.edu.pe/noticias-y-eventos/noticias/movilidad-sostenible-a-que-desafios-y-problemas-se-enfrenta-lima-metropolitana/>
- [10] NFC Forum, NFC. 2006. <http://www.nfc-forum.org/aboutnfc> visitada en Julio 2006.
- [11] Chavarria, D., "Título de la tabla," en *Tecnología de comunicación de campo cercano (NFC) y sus aplicaciones*, Trabajo de Grado, Universidad de Costa Rica, 2011.
- [12] R. M. Avilés, "Ventajas de la tecnología Near Field Communication (NFC) como sistema de pago electrónico," *RECI Revista Iberoamericana de las Ciencias Computacionales e Informática*, 2022. [En línea]. Disponible: <https://www.reci.org.mx/index.php/reci/article/download/55/256?inline=1>. [Accedido: 09- Nov- 2023].
- [13] Edgar Serna M. (Ed.). *Métodos Formales, Ingeniería de Requisitos y Pruebas del Software*. 2021 Medellín - Antioquia
- [14] E. Vidal-Duarte, C. Mogrovejo Ramirez, y E. Castro Gutierrez, "Reduciendo la Ambigüedad en el Modelo del Dominio mediante Especificaciones Formales Ligeras en VDM++," *Universidad Católica San Pablo, Universidad Nacional de San Agustín*, 2011. [En línea]. Disponible: <http://repositorio.uigv.edu.pe/bitstream/handle/20.500.11818/896/COMTEL-2011-30-33.pdf?sequence=1>
- [15] John Fitzgerald, Peter Gorm Larsen, Paul Mukherjee, Nico Plat, Marcel Verhoef. *Validated Designs for Object oriented Systems*. Version 1.2. Springer. September 22nd 2004.
- [16] VDMTools. *The VDM++ Language Manual 2.0*. Revised for VDMTools v9.0.6. COPYRIGHT 2016 by Kyushu University. <http://fmvdm.org/doc/index.html>
- [17] CSK Systems Corporation, "The VDM++ Language Manual", 2009. <http://fmvdm.org/doc/index.html>
- [18] H. Chacca, R. Montufar, R. Zenteno, E. Vidal, "Uso de Especificaciones Formales con VDM en el Acceso de Usuarios en una Red Social Universitaria: Caso Uconecta," *Universidad Nacional San Agustín, Perú*, 2017. [En línea]. Disponible: [https://www.lacpei.org/LACPEI2017-BocaRaton/student\\_Papers/SP229.pdf](https://www.lacpei.org/LACPEI2017-BocaRaton/student_Papers/SP229.pdf)



# Gracias