**WebService creados por Bonus**

**1.      Logeo de usuario.**

Ruta: [**http://200.62.147.188:8080/app/servlet/awslogusugx**](http://200.62.147.188:8080/app/servlet/awslogusugx)[**?wsdl**](http://200.62.147.188:8080/app/servlet/awslisdis)

Parámetros de entrada: **In:&UsuCod C(10), In:&UsuCla C(10)**

Parámetros de salida: **Out:&CodError N(4), Out:&MsjError C(100)**

Mapeo de errores:

**0: Éxito**

**1: Usuario en blanco**

**2: Clave en blanco**

**3: Usuario no existe**

**4: Usuario no pertenece al grupo autorizado**

**5: Clave incorrecta**

Nota: Utilizaremos el usuario: USUAPP1 y clave: bonus para las pruebas.  
**WS en Valmar:** [**POST authenticate**](http://54.191.211.214/wsBonus/Help/Api/POST-authenticate)

**2.      Lista de código y descripción de tipo de documento.**

Ruta:[**http://200.62.147.188:8080/app/servlet/awslistipdo**](http://200.62.147.188:8080/app/servlet/awslistipdo)[**?wsdl**](http://200.62.147.188:8080/app/servlet/awslisdis)

Parámetros de salida: **Out:&Lista(Colección) {Out:Lista.TipDocCod N(4), Out:Lista.TipDocNom C(20)}**

Mapeo de errores:

**Algún error trae lista vacía**

**WS en Valmar:** [**GET ObtenerDocumentos**](http://54.191.211.214/wsBonus/Help/Api/GET-ObtenerDocumentos)

**3.      Lista de código y descripción de departamentos.**

**Ruta: http://200.62.147.188:8080/app/servlet/awslisdep?wsdl**

**Parámetros de salida: Out:&Lista(Colección) {Out:Lista.DptoCod C(3), Out:Lista.DptoDes C(25)}**

**Mapeo de errores:**

**Algún error trae lista vacía**

**WS en Valmar:** [**GET ObtenerDepartamentos**](http://54.191.211.214/wsBonus/Help/Api/GET-ObtenerDepartamentos)

**4.      Lista de código y descripción de provincias.**

Ruta:[**http://200.62.147.188:8080/app/servlet/awslispro**](http://200.62.147.188:8080/app/servlet/awslispro)[**?wsdl**](http://200.62.147.188:8080/app/servlet/awslisdis)

Parámetros de entrada: **In:&DptoCod**

Parámetros de salida: **Out:&Lista(Colección) {Out:Lista.ProvCod C(3), Out:Lista.ProvDes C(25)}**

**Mapeo de errores:**

**Algún error trae lista vacía**

**WS en Valmar:** [**GET ObtenerProvincias**](http://54.191.211.214/wsBonus/Help/Api/GET-ObtenerProvincias_dptoCod)

**5.      Lista de código y descripción de distritos.**

Ruta:[**http://200.62.147.188:8080/app/servlet/awslisdis?wsdl**](http://200.62.147.188:8080/app/servlet/awslisdis)

Parámetros de entrada: **In:&DptoCod, In:&ProvCod**

Parámetros de salida: **Out:&Lista(Colección) {Out:Lista.DistCod C(3), Out:Lista.DistDes C(25), Out:Lista.DistCodPos C(6)}**

**Mapeo de errores:**

**Algún error trae lista vacía**  
 **WS en Valmar:** [**GET ObtenerDistritos**](http://54.191.211.214/wsBonus/Help/Api/GET-ObtenerDistritos_dptoCod_provCod)

**1.       Insertar token.**

**Ruta: h**[**ttp://200.62.147.188:8080/app/servlet/awsinstoken?wsdl**](http://200.62.147.188:8080/app/servlet/awsinstoken?wsdl)

**Parámetros de entrada: In:&UserId C(10), In:&AuthToken C(20), In:&IssuedOn T(8.5), In:&ExpiresOn T(8.5)**

**Parámetros de salida: Out:&CodError N(4), Out:&MsjError C(100)**

**Mapeo de errores:**

**0: Éxito**

**1: UserId en blanco**

**2: AuthToken en blanco**

**3: IssuedOn en blanco**

**4: ExpiresOn en blanco**

**5: TokenId duplicada**

**WS en Valamr: utlizado en WS Autenticacion**

**2.       Obtener token.**

**Ruta:** [**http://200.62.147.188:8080/app/servlet/awsobttoken?wsdl**](http://200.62.147.188:8080/app/servlet/awsobttoken?wsdl)

**Parámetros de entrada: In:&AuthToken C(20), In:&CurrentDate T(8.5)**

**Parámetros de salida: Out:&CodError N(4), Out:&MsjError C(100), Out:&TokenId N(6), Out:&UserId C(10), Out:&IssuedOn T(8.5), Out:&ExpiresOn T(8.5)**

**Mapeo de errores:**

**0: Éxito**

**1: AuthToken en blanco**

**2: CurrentDate en blanco**

**3: No hay registros**

**WS en Valamr: utlizado en WS Autenticacion**

**3.       Actualizar token.**

**Ruta:** [**http://200.62.147.188:8080/app/servlet/awsacttoken?wsdl**](http://200.62.147.188:8080/app/servlet/awsacttoken?wsdl)

**Parámetros de entrada: In:&AuthToken C(20), In:& ExpiresOn T(8.5)**

**Parámetros de salida: Out:&CodError N(4), Out:&MsjError C(100)**

**Mapeo de errores:**

**0: Éxito**

**1: AuthToken en blanco**

**2: ExpiresOn en blanco**

**3: No se encontró AuthToken ingresado**

**WS en Valamr: utlizado en WS Autenticacion**

**4.       Eliminar token.**

**Ruta:** [**http://200.62.147.188:8080/app/servlet/awselitoken?wsdl**](http://200.62.147.188:8080/app/servlet/awselitoken?wsdl)

**Parámetros de entrada: In:&AuthToken C(20)**

**Parámetros de salida: Out:&CodError N(4), Out:&MsjError C(100)**

**Mapeo de errores:**

**0: Éxito**

**1: AuthToken en blanco**

**2: No se encontró AuthToken ingresado**

**WS en Valamr: utlizado en WS Autenticacion**

**5.       Eliminar token por usuario.**

**Ruta:** [**http://200.62.147.188:8080/app/servlet/awselitokus?wsdl**](http://200.62.147.188:8080/app/servlet/awselitokus?wsdl)

**Parámetros de entrada: In:&UserId C(10)**

**Parámetros de salida: Out:&CodError N(4), Out:&MsjError C(100)**

**Mapeo de errores:**

**0: Éxito**

**1: UserId en blanco**

**2: No se encontró UserId ingresado**

**WS en Valamr: utlizado en WS Autenticacion**

**6.**       **Count token.**

**Ruta:** [**http://200.62.147.188:8080/app/servlet/awsobtcotok?wsdl**](http://200.62.147.188:8080/app/servlet/awsobtcotok?wsdl)

**de entrada: In:&AuthToken C(20)**

**Parámetros de salida: Out:&Count N(4), Out:&CodError N(4), Out:&MsjError C(100)**

**Mapeo de errores:**

**0: Éxito**

**1: AuthToken en blanco**

**2: No se encontró AuthToken ingresado**

**WS en Valamr: utlizado en WS Autenticacion**

**7.**       **Count token por usuario.**

**Ruta:** [**http://200.62.147.188:8080/app/servlet/awsobtctous?wsdl**](http://200.62.147.188:8080/app/servlet/awsobtctous?wsdl)

**Parámetros de entrada: In:&UserId C(10)**

**Parámetros de salida: Out:&Count N(4), Out:&CodError N(4), Out:&MsjError C(100)**

**Mapeo de errores:**

**0: Éxito**

**1: UserId en blanco**

**2: No se encontró UserId ingresado**

**WS en Valamr: utlizado en WS Autenticacion**

**1.**       **Existe Cliente Bonus.**

Ruta:[**http://200.62.147.188:8080/app/servlet/awsexiclibo?wsdl**](http://200.62.147.188:8080/app/servlet/awsexiclibo?wsdl)

de entrada: **In: &TipDocCod N(4), In: &PrsNroDoc C(15)**

Parámetros de salida: **Out:&CodError N(4), Out:&MsjError C(100), Out:&PrsCod C(10)**

Mapeo de errores:

**0: Éxito**

**1: Código de tipo de documento nulo**

**2: Número de documento nulo**

**3: Tipo y número de documento no encontrado**

**4: Código de persona nula**

**5: No existe cliente bonus**

**Nota: Datos de prueba TipDocCod = 1, PrsNroDoc=’** **00000008’**

**WS en Valmar:** [**GET ExisteCliente**](http://54.191.211.214/wsBonus/Help/Api/GET-ExisteCliente_tipoDocCod_prsNroDoc)

**2.**       **Devuelve datos principales y secundarios de cliente (completo).**

Ruta:[**http://200.62.147.188:8080/app/servlet/awsdevdatco?wsdl**](http://200.62.147.188:8080/app/servlet/awsdevdatco?wsdl)

Parámetros de entrada: **In:&PrsCod C(10)**

Parámetros de salida: **Out:&CodError N(4), Out:&MsjError C(100), Out:&PrsApePat C(20), Out:&PrsApeMat C(20), Out:&PrsPriNom C(15), Out:&PrsSegNom C(15), Out:&PrsTerNom C(15), Out:&TipDocCod N(1), Out:&PrsNroDoc C(15), Out:&PrsSex C(1), Out:&PrsFecNac D(8), Out:&Texto1 C(50), Out:&Texto2 C(50), Out:&Texto3 C(50), Out:&Texto4 C(50), Out:&Texto5 C(50), Out:&CarritoCo (Colección) {Out:CarritoCo.PrsMaiCod N(2), Out:CarritoCo.PrsMai C(40),Out: CarritoCo.PrsFlgMErr C(1)}, Out:&CarritoTe (Colección) {Out:&CarritoTe.PrsTlfCod N(2), Out:&CarritoTe.PrsPreTlf N(5), Out:&CarritoTe.PrsNroTlf N(10), Out:&CarritoTe.PrsAnxTlf N(5), Out:&CarritoTe.PrsFlgMov N(1), Out:&CarritoTe.PrsRedPrv C(20), Out:&CarritoTe.TlfRefCod N(2), Out:&CarritoTe.OtfCod N(4), Out:&CarritoTe.TlfRefCod N(2), Out:&CarritoTe.PrsFlgTErr C(1)}, Out:&Direccion C(200), Out:&Referencia C(150), Out:&DptoCod C(2), Out:&ProvCod C(2), Out:&DistCod C(2), Out:&FlgTieVeh C(1), Out:&FlgTieHij C(1), Out:&CarritoHij (Colección) {Out:&CarritoHij.PrsHijEda N(3), Out:&CarritoHij.PrsHijSex C(1)}**

Mapeo de errores:

**0: Éxito**

**1: Código de persona nula**

**2: Código de persona no existe**

**WS en Valmar:** [**GET Cliente**](http://54.191.211.214/wsBonus/Help/Api/GET-Cliente_prsCod)

**1. Lista de n últimos movimientos de fidelización (puntos)**

Ruta: <http://200.62.147.188:8080/app/servlet/awsultmovpt?wsdl>

Parámetros de entrada: **In:&CtaPrsCod C(10), In:&CtaCod N(3)**

Parámetros de salida: **Out:&CodError N(4), Out:&MsjError C(100), Out:&NroTrn N(5), Out:&Transaccion (Colección) {Out:&Transaccion.FchHor C(15), Out:&Transaccion.HorProc C(8), Out:&Transaccion.FchProc C(10), Out:&Transaccion.FchAsig C(10), Out:&Transaccion.Descrip C(60), Out:&Transaccion.PtosAsig N(16.2)-, Out:&Transaccion.PtosCanj N(16.2)-, Out:&Transaccion.EsCanje C(1)}**

Mapeo de errores:

**0: Éxito**

**1: Código de persona nula**

**2: Código de cuenta nula**

**3: Código de persona no existe**

**Nota: Datos de prueba TipDocCod = 1, PrsNroDoc=’0001374662’**

**WS de Valmar:** [**GET ObtenerMovFidelizacion**](http://54.191.211.214/wsBonus/Help/Api/GET-ObtenerMovFidelizacion_ctaPrsCod_ctaCod)

**1. Lista de número y tipo de cuenta, nombre del titular, nombre del tarjetahabiente y saldo de fidelización (puntos) por cuenta.**

Ruta: <http://200.62.147.188:8080/app/servlet/awslisctapt?wsdl>

de entrada: **In:&CtaPrsCod C(10)**

Parámetros de salida: **Out:&CodError N(4), Out:&MsjError C(100), Out:&CarritoCta (Colección), {Out:CarritoCta.CtaPrsCod C(10), Out:CarritoCta.CtaCod N(3), Out: CarritoCta.CtaPrsNom C(60), Out: CarritoCta.PCtaAsoCod C(10), Out: CarritoCta.CtaAsoNom C(60), Out: CarritoCta.PCtaTip C(1), Out: CarritoCta.PCtaAutCnj C(1), Out: CarritoCta.TipPunCod N(4), Out: CarritoCta.CtaSalVig N(16.2)-, Out: CarritoCta.CtaSalCon N(16.2)-, Out: CarritoCta.CtaSalDsp N(16.2)-}**

Mapeo de errores:

**0: Éxito**

**1: Código de persona nula**

**2: Código de persona no existe**

**Nota: Datos de prueba PrsNroDoc= ’0001374662’**

**WS en Valmar:** [**GET ObtenerCuentas**](http://54.191.211.214/wsBonus/Help/Api/GET-ObtenerCuentas_ctaPrsCod)