

Cobranza Móvil DEVOPS – FLUJO CI/CD

¿Qué tenemos hoy?

Ambiente local de desarrollo

- Repositorio de código.
- Proceso de automatización de actividades.
- Repositorio de artefactos.
- Revisión de código estático.
- Revisión de código seguro.
- Pruebas de integración.
- Reportes y métricas de calidad del código.

Ambientes de pruebas continuas

- Ambientes para la CNBV -> DEV, INT, QA.
- Procesos de automatización con pipelines Git->Build->Deploy->Test.
- Pruebas de Integración.
- Reportes y métricas de calidad del código.

Ambiente productivo

- Proceso de automatización para liberación de cambios.
- Repositorio de código corporativo.
- Revisión de código estático.
- Repositorio de artefactos corporativo.
- Reportes y métricas de calidad del código.



Jenkins



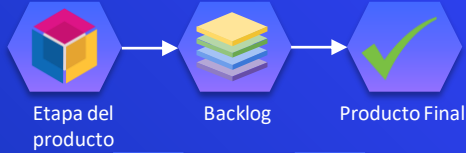
sonarqube

Checkmarx



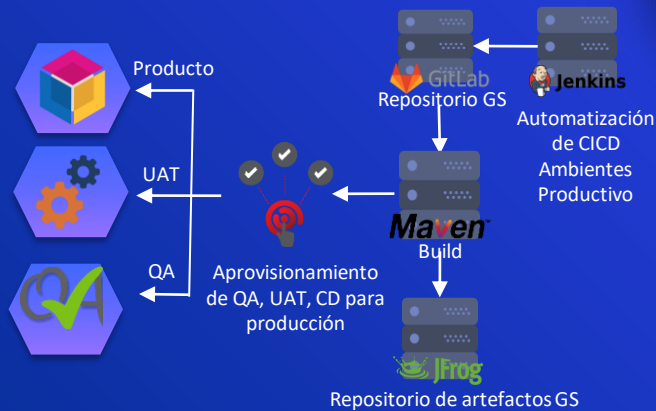
DESARROLLO ÁGIL

Standup Diario

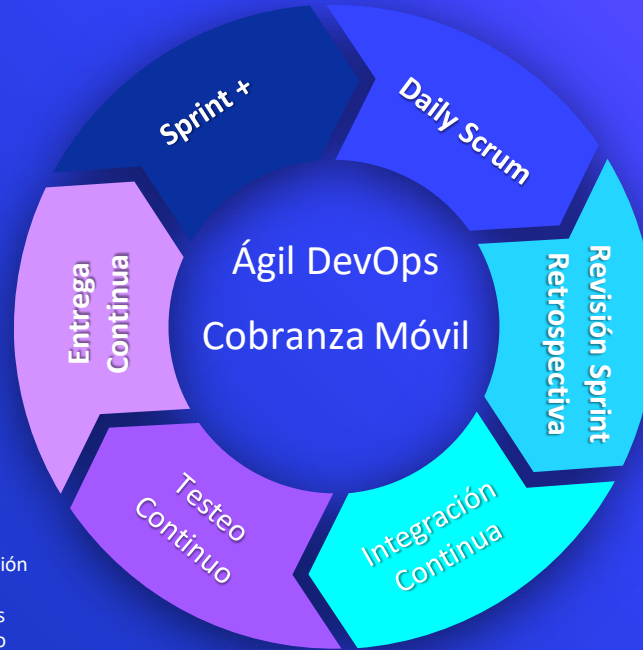


Feedback continuo

DESPLIEGUE CONTINUO

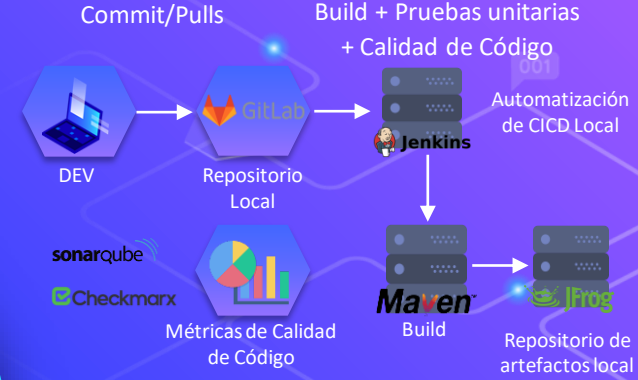


Feedback continuo



Feedback continuo

INTEGRACIÓN CONTINUA



Feedback continuo

TESTEO CONTINUO



Arquitectura Cobranza Móvil DEV CNBV



IEECO Ecosistema:441
IEECO Aplicación:1507



FQDN: dev.cbm.apps.cbz.baz.cloud
IP: 10.82.56.253
Puerto: 8444
HTTPS: TLS 1.2



1 VCU
RAM: 4 GB
Cores: 4
HD: 100 GB
Puertos: 8444,8445
IEECO: 6294,6295



Web Server APP: JBOSS 7.3
SO: RedHat Linux 7.8
Java Versión: 1.8



S.O: Windows Server 2016 Datacenter
10.0
Motor: Microsoft SQL Server 2019
Enterprise Edition x64 Bits
Cores: 8
RAM: 16 GB
HD: 200 GB
Puerto: 1533
IEECO: 6046
IEECODB: 936



Microsoft
SQL Server
2019

Arquitectura Cobranza Móvil INT CNBV



IEECO Ecosistema:442
IEECO Aplicación:1508



FQDN: int.cbm.apps.cbz.baz.cloud
IP: 10.82.58.253
Puerto: 8444
HTTPS: TLS 1.2



1 VCU
RAM: 4 GB
Cores: 4
HD: 100 GB
Puertos: 8444,8445
IEECO: 6296, 6297



Web Server APP: JBOSS 7.3
SO: RedHat Linux 7.8
Java Versión: 1.8



S.O: Windows Server 2016 Datacenter 10.0
Motor: Microsoft SQL Server 2019 Enterprise Edition x64 Bits
Cores: 8
RAM: 16 GB
HD: 200 GB
Puerto: 1533
IEECO: 6047
IEECOBD: 938



Microsoft
SQL Server
2019

Arquitectura Cobranza Móvil QA CNBV



IEECO Ecosistema:443
IEECO Aplicación:1509



FQDN: qa.cbm.apps.cbz.baz.cloud
IP: 10.82.54.253
Puerto: 8444
HTTPS: TLS 1.2



1 VCU
RAM: 4 GB
Cores: 4
HD: 100 GB
Puertos: 8444,8445
IEECO: 6298,9299



Web Server APP: JBOSS 7.3
SO: RedHat Linux 7.8
Java Versión: 1.8

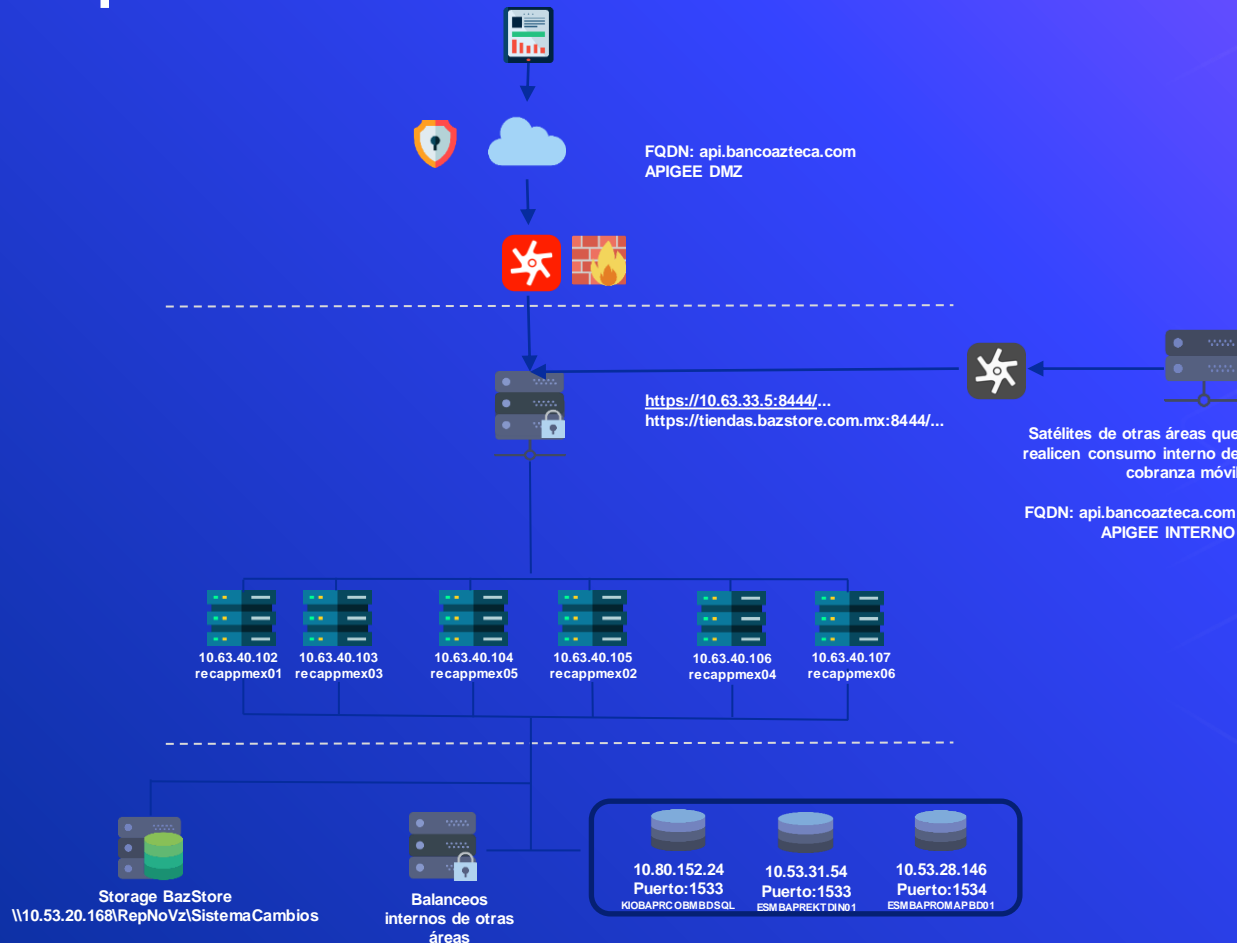


S.O: Windows Server 2016 Datacenter
10.0
Motor: Microsoft SQL Server 2019
Enterprise Edition x64 Bits
Cores: 8
RAM: 16 GB
HD: 200 GB
Puerto: 1533
IEECO: 6048
IEECOBD: 938



SQL Server
2019

Arquitectura Cobranza Móvil CDMX PROD



IEECO Ecosistema:19
IEECO Aplicación:76



APIGEE on premise para internet
Dominio: api.bancoazteca.com
Uso: Consumo de servicios para DMZ
desde dispositivos móviles.



APIGEE on premise interno
Uso: Consumo de servicios internos configurado por apigee



Balanceo Interno
IP: 10.63.33.5
Puerto: 8444
FQDN: tiendas.bazstore.com.mx



6 VCU
RAM: 24 GB
Cores: 8
HD: 50 GB
Puertos: 8444,8445
IEECO: 986,987,988,989,990,991

Web Server APP: JBOSS 7.2
SO: RedHat Linux 7.7
Java Versión: 8



S.O: Windows Server 2019 Datacenter
10.0
Motor: Microsoft SQL Server 2019
Enterprise Edition x64 Bits
IEECO:3706,3707,3708,3709



Arquitectura Cobranza Móvil KIO PROD



IEECO :202
IEECO ApEcosistemallicación:1033



APIGEE on premise para internet
Dominio: `api.bancoazteca.com`
Uso: Consumo de servicios para DMZ desde dispositivos móviles.



APIGEE on premise interno
Uso: Consumo de servicios internos configurado por apigee



Balaneo Interno
IP: `10.80.150.98`
Puerto: `8444`
FQDN: `cbm.apps.cbz.baz.cloud`



6 VCU
RAM: 24 GB
Cores: 8
HD: 40 GB
Puestos: `8444,8445`
IEECO: `3710,3711,3712,3713,3714,3715`



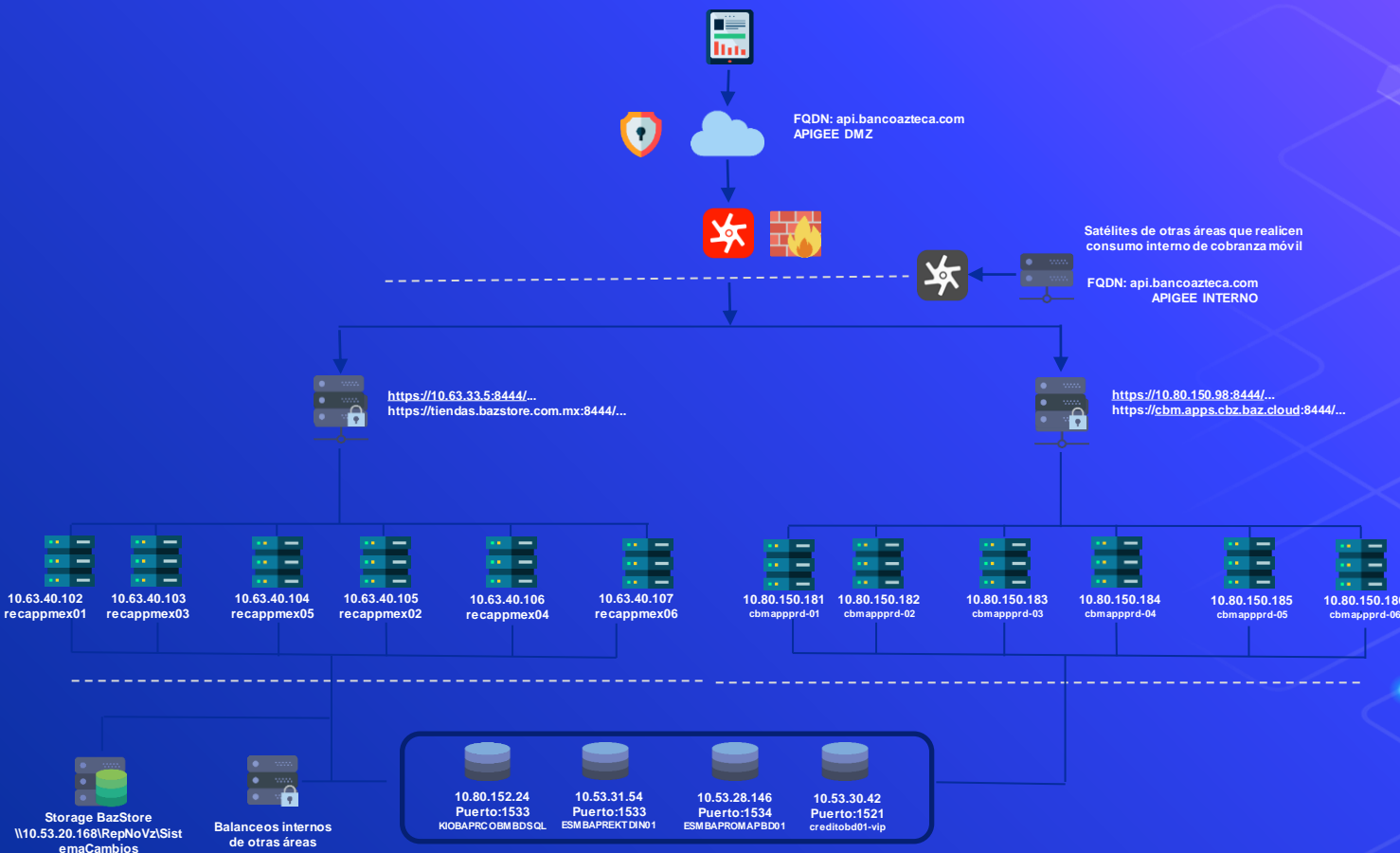
Web Server APP: JBOSS 7.3
SO: RedHat Linux 8.1
Java Versión: 8



S.O:

Windows Server 2019 Datacenter 10.0
Motor: Microsoft SQL Server 2019 Enterprise Edition x64 Bits
IEECO: `3706,3707,3708,3709`

Arquitectura Cobranza Móvil



APIGEE on premise para internet
Dominio: api.bancoazteca.com
Uso: Consumo de servicios para DMZ desde dispositivos móviles.



APIGEE on premise interno
Uso: Consumo de servicios internos configurado por apigee

Cobranza Movil Kio

IEECO Ecosistema:202 Aplicación:1033



Balaneo Interno
IP: 10.80.150.98 Puerto: 8444
FQDN: cbm.apps.cbz.baz.cloud



6 VCU RAM: 24 GB Cores: 8
HD: 40 GB Puertos: 8444,8445
IEECO:3710,3711,3712,3713,3714,3715



Web Server APP: JBOSS 7.3
SO: RedHat Linux 8.1
Java Versión: 8



S.O: Windows Server 2019 Datacenter 10.0
Motor: Microsoft SQL Server 2019 Enterprise Edition x64 Bits
IEECO:3709

Cobranza Movil CDMX

IEECO Ecosistema:19 Aplicación:76



Balaneo Interno
IP: 10.63.33.5 Puerto: 8444
FQDN: tiendas.bazstore.com.mx



6 VCU RAM: 24 GB Cores: 8
HD: 50 GB Puertos: 8444,8445
IEECO: 986,987,988,989,990,991



Web Server APP: JBOSS 7.2
SO: RedHat Linux 7.7
Java Versión: 8



S.O: Windows Server 2019 Datacenter 10.0
Motor: Microsoft SQL Server 2019 Enterprise Edition x64 Bits
IEECO:3709

Arquitectura Cobranza Móvil Quartz



Beneficios de implementar DevOps



30% Reducción

En tiempo sobre mantenimientos y correcciones de aplicaciones.



25% Mejora

En el desempeño y calidad de las aplicaciones.



20% Incremento

En ganancias al reducir los tiempos de desarrollo, pruebas, despliegues. Con garantía de Calidad.



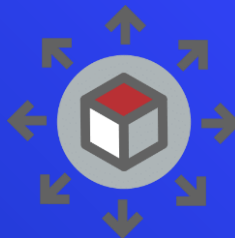
25% Incremento

En tiempo para mejoramientos sobre la infraestructura para DevOps.



30% Incremento

En tiempo para el autoaprendizaje sobre nuevas tecnologías y soluciones.



60% Decremento

En el tiempo para liberar nuevos productos.



Cobertura del 100%?
Cómo alcanzar la meta.



Nuestro flujo



Gestión de requerimiento y OT



Checkout y desarrollo de código

Maven

Build de la aplicación



Pruebas unitarias

Commits/Pulls repositorio local



Jenkins

Trigger procesos de automatización

Obtención últimos cambios



Respaldo de artefacto



Pruebas de integración



Deploy en ambiente local de desarrollo



Revisión de código seguro



Revisión código estático

Build de la aplicación



Jenkins

Trigger ambientes de pruebas continuas Dev, Int, QA

Obtención de últimos cambios

Build de la aplicación para cada ambiente

Deploy de la aplicación para cada ambiente

Pruebas de integración por cada ambiente

Actualización de OT documentación de cambio

Nuestro flujo



Actualización
OT a
desarrollado



Respaldo de
artefacto GS



Revisión de
código estático
GS



Build de la
aplicación GS



Commits/Pulls
repositorio GS



Trigger proceso de
automatización para
liberación de cambio



Pruebas de QA

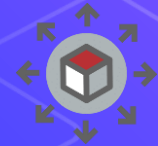


Aprovisionamiento
de resultados



Cambio de estatus
OT Probado

Agenda de cambio
por medio de CD



Solicitud de firmas
y actualización de
fecha para
liberación

Despliegue del
cambio en
producción



Entrega del
producto



Actualización de
OT a Cerrado

Integración de
código a Master
en Repositorio
de código de GS

Actualización de
OT a
implementado

Confirmación y
autorización para
dejar desplegado
el cambio

Pruebas y Validación de
cambio aplicado




```

1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5
6  my $script = $0;
7  my $version = "1.0.0";
8  my $author = "John Doe";
9  my $email = "john.doe@example.com";
10
11  my $help = "
12  Usage: perl $script [options]
13
14  Options:
15  -h, --help            Display this help message
16  -v, --version          Display the version number
17  -a, --author           Display the author's name
18  -e, --email            Display the author's email address
19  -f, --file FILE        Process the file specified by FILE
20  -o, --output FILE      Write the output to the file specified by FILE
21  -s, --size SIZE        Set the size of the output file to SIZE
22  -t, --type TYPE        Set the type of the output file to TYPE
23  -m, --mode MODE        Set the mode of the output file to MODE
24  -p, --pattern PATTERN  Set the pattern to match in the file
25  -c, --count COUNT      Set the count of the output file to COUNT
26  -d, --debug            Enable debug mode
27  -q, --quiet            Suppress all output
28  -v, --verbose           Enable verbose output
29  -h, --help            Display this help message
30
31  Examples:
32  perl $script -f input.txt -o output.txt
33  perl $script -f input.txt -o output.txt -s 1024 -t text -m 0644
34  perl $script -f input.txt -o output.txt -p 'pattern' -c 100
35  perl $script -f input.txt -o output.txt -d
36  perl $script -f input.txt -o output.txt -q
37  perl $script -f input.txt -o output.txt -v
38
39  Copyright (c) 2023 John Doe. All rights reserved.
40  License: MIT
41  "
42
43  my $argc = scalar(@ARGV);
44  my $argv = join(" ", @ARGV);
45
46  if ($argc == 0) {
47      print "No arguments provided. Please use the help option for more information.\n";
48      exit 1;
49  }
50
51  my $file = "";
52  my $output = "";
53  my $size = 1024;
54  my $type = "text";
55  my $mode = 0644;
56  my $pattern = "";
57  my $count = 100;
58  my $debug = 0;
59  my $quiet = 0;
60  my $verbose = 0;
61
62  for (my $i = 0; $i < $argc; $i++) {
63      my $arg = $ARGV[$i];
64      my $value = "";
65
66      if ($arg =~ /^-h$/) {
67          print $help;
68          exit 0;
69      }
70      elsif ($arg =~ /^-v$/) {
69          print $version;
70          exit 0;
71      }
72      elsif ($arg =~ /^-a$/) {
73          print $author;
74          exit 0;
75      }
76      elsif ($arg =~ /^-e$/) {
77          print $email;
78          exit 0;
79      }
80      elsif ($arg =~ /^-f$/) {
81          $file = $ARGV[$i+1];
82          $i++;
83      }
84      elsif ($arg =~ /^-o$/) {
85          $output = $ARGV[$i+1];
86          $i++;
87      }
88      elsif ($arg =~ /^-s$/) {
89          $size = $ARGV[$i+1];
90          $i++;
91      }
92      elsif ($arg =~ /^-t$/) {
93          $type = $ARGV[$i+1];
94          $i++;
95      }
96      elsif ($arg =~ /^-m$/) {
97          $mode = $ARGV[$i+1];
98          $i++;
99      }
100     elsif ($arg =~ /^-p$/) {
101         $pattern = $ARGV[$i+1];
102         $i++;
103     }
104     elsif ($arg =~ /^-c$/) {
105         $count = $ARGV[$i+1];
106         $i++;
107     }
108     elsif ($arg =~ /^-d$/) {
109         $debug = 1;
110     }
111     elsif ($arg =~ /^-q$/) {
112         $quiet = 1;
113     }
114     elsif ($arg =~ /^-v$/) {
115         $verbose = 1;
116     }
117 }
118
119 if ($file == "") {
120     print "Error: No file specified. Please use the -f option to specify the input file.\n";
121     exit 1;
122 }
123
124 if ($output == "") {
125     print "Error: No output file specified. Please use the -o option to specify the output file.\n";
126     exit 1;
127 }
128
129 if ($size == 0) {
130     print "Error: Invalid size. Please use the -s option to specify a valid size.\n";
131     exit 1;
132 }
133
134 if ($type == "") {
135     print "Error: Invalid type. Please use the -t option to specify a valid type.\n";
136     exit 1;
137 }
138
139 if ($mode == 0) {
140     print "Error: Invalid mode. Please use the -m option to specify a valid mode.\n";
141     exit 1;
142 }
143
144 if ($pattern == "") {
145     print "Error: Invalid pattern. Please use the -p option to specify a valid pattern.\n";
146     exit 1;
147 }
148
149 if ($count == 0) {
150     print "Error: Invalid count. Please use the -c option to specify a valid count.\n";
151     exit 1;
152 }
153
154 if ($debug == 1) {
155     print "Debug mode enabled.\n";
156 }
157
158 if ($quiet == 1) {
159     print "Quiet mode enabled.\n";
160 }
161
162 if ($verbose == 1) {
163     print "Verbose mode enabled.\n";
164 }
165
166 # Main processing logic
167 # ...
168
169 # End of script

```

[illegible]

The screenshot shows the CPLEX 12.6.3.0 interface. The top bar includes 'File', 'Solve', 'Tools', 'Help', and 'Log'. The main window displays the 'Problem Explorer' on the left with a tree view showing 'Model' and 'Solve'. The central area shows the 'Model Explorer' with a list of variables and constraints. The bottom status bar indicates 'Solving' and '1.0000000000000000'.

[illegible]



Cambios totos para ser integrados?



Estado: **Exito**

ID: cym2000000
 Fecha: April 17
 Estado: PENDING
 ID: 200000
 Responsable: EDGAR HERRERA GARCIA















Crea tu perfil y comienza a recibir ofertas de trabajo en tu perfil.

[Comenzar a usar Simba](#)

The screenshot shows the Bitly dashboard with the 'Exit' button highlighted in green. A large green checkmark icon is overlaid on the right side of the dashboard.

[illegible]


Banco Azteca

[illegible]

Solos no llegamos lejos, trabajando juntos no tenemos límites.

Colaboradores

Cobranza Relacional – Desarrollo

QA Central - Pruebas

DSI – Total Sec – Labs Tools

ACC - Control de Versiones

ACC - Integración Continua

Operaciones Cloud - Despliegue Continuo

Operaciones Unix – Producción Server Apps

Operaciones Unix DRP – Ambientes CNBV Dev,Int,QA Server Apps

DBA Alnova SQL – Desarrollo/Producción

Monitoreo de Operaciones y Aplicativo

CMASADMIN



Gracias por su apoyo



Cobranza Relacional

Dirección de Cobranza y Crédito