In[93]:=
```
ClearAll["Global`*"]
```

In[1]:=
```
TwoAxisPlot[{f_, g_}, {x_, x1_, x2_}] := Module[
  {fgraph, ggraph, frange, grange, fticks, gticks}, {fgraph, ggraph} = MapIndexed[
    Plot[#, {x, x1, x2}, Axes → True, PlotStyle → ColorData[1][#2[[1]]]] &, {f, g}];
  {frange, grange} = (PlotRange /. AbsoluteOptions[#, PlotRange])[[2]] & /@
    {fgraph, ggraph};
  fticks = N@FindDivisions[frange, 5];
  gticks = Quiet@Transpose@{fticks, ToString[NumberForm[#, 2], StandardForm] & /@
       Rescale[fticks, frange, grange]};
  Show[fgraph, ggraph /. Graphics[graph_, s___] :>
     Graphics[GeometricTransformation[graph,
       RescalingTransform[{{0, 1}, grange}, {{0, 1}, frange}]], s], Axes → False,
   Frame → True, FrameStyle → {ColorData[1] /@ {1, 2}, {Automatic, Automatic}},
   FrameTicks → {{fticks, gticks}, {Automatic, Automatic}}]]
```

In[3]:=

Practica 1 Redes neuronales

In[63]:=
```
Needs["PhysicalConstants`"]
```

In[64]:=
```
Temperature = 273 + 20;
```

Ejercicio 1:

Usando la ecuacion de Nerst, determinar los potenciales de equilibrio para los siguientes iones

Defino la ecuación de Nerst

In[65]:=
```
BoltzConst = 8.6 * 10 ^ -5;
```

In[66]:=
```
VEquil[cOut_, cIn_] := N[ BoltzConst * Temperature Log[cOut / cIn], 2]
```

In[67]:=
```
values = Association["K" -> {430, 20}, "Na" -> {50, 440}, "Cl" -> {65, 550}];
```

In[68]:=
```
(VEquil[#[[1]], #[[2]]]) & /@ values
```

Out[68]=
```
<|K → 0.0773088, Na → -0.0547994, Cl → -0.0538111|>
```

Ejercicio 2

Considerar una neurona esférica con un radio de 15 micrones y una capacitancia de 1 F / cm^2. Que cantidad de iones de sodio deben ingresar a la neurona para cambiar el potencial de membrana en 100 mV? Comparar el cambio de concentración con la concentración de iones de sodio del problema anterior. Ayuda: usar como valor de la constante de Faraday: $F = 10^5$ Coulombs/mol

In[94]:=
```
Module[{r, C, V, F, A, n},
 r = 15*^-3;
 C = 1*^-3/(1*^-2)^2;
 V = 100*^-3;
 F = 1*^5;
 A = 4 * Pi * r^2;
 n = N[C V A/F]
]
```

Out[94]= 2.82743×10^{-8}

Este es el valor de concentracion necesario, se ve que es mucho menor que

Ejercicio 3

Simular la dinámica de una neurona de Hogdkin-Huxley

In[2]:=

$$a_m[V\_] := 0.1 \frac{(V + 40)}{1 - \mathrm{Exp}\left[-\frac{V+40}{10}\right]}$$

$$b_m[V\_] := 4\,\mathrm{Exp}[-(V + 65)/18]$$

$$a_h[V\_] := 0.07\,\mathrm{Exp}[-(V + 65)/20]$$

$$b_h[V\_] := \frac{1}{1 + \mathrm{Exp}[-(V + 35)/10]}$$

$$a_n[V\_] := 0.01 \frac{(V + 55)}{1 - \mathrm{Exp}[-(V + 55)/10]}$$

$$b_n[V\_] := 0.125\,\mathrm{Exp}[-(V + 65)/80]$$

In[8]:=

$$m_\infty[V\_] := a_m[V]/(a_m[V] + b_m[V])$$

$$h_\infty[V\_] := a_h[V]/(a_h[V] + b_h[V])$$

$$n_\infty[V\_] := a_n[V]/(a_n[V] + b_n[V])$$

$$tau_m[V\_] := 1/(a_m[V] + b_m[V])$$

$$tau_h[V\_] := 1/(a_h[V] + b_h[V])$$

$$tau_n[V\_] := 1/(a_n[V] + b_n[V])$$

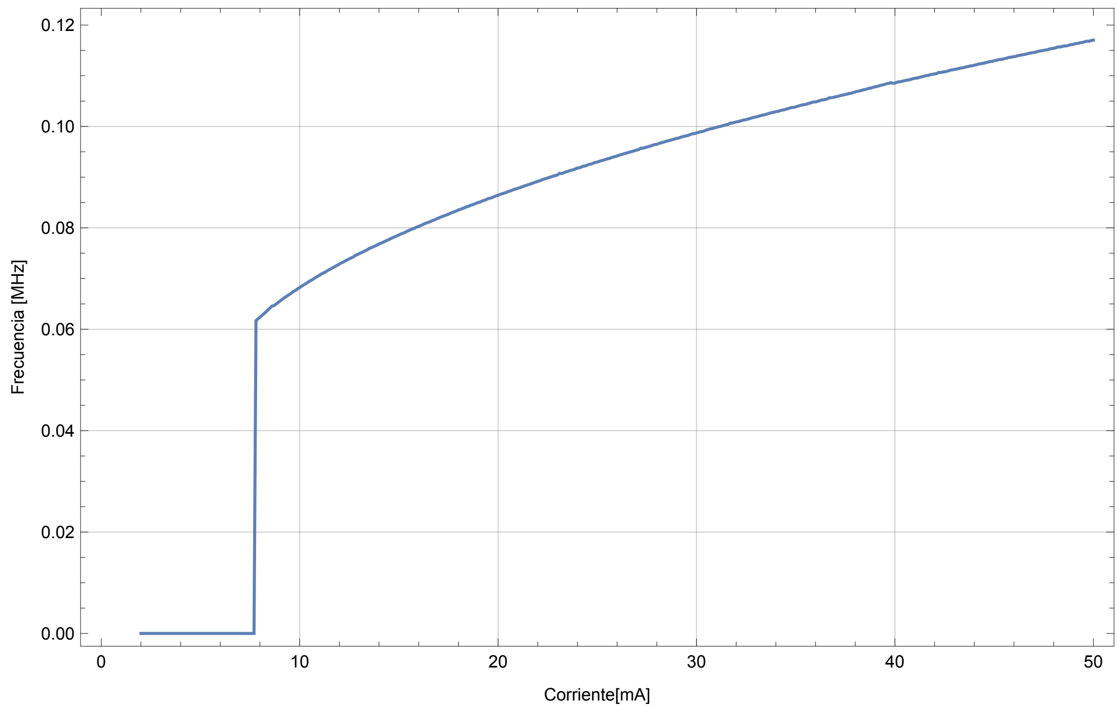
Inicializo los parametros externos

```
In[14]:= Cap = 1;
VNa = 50;
Vk = -77;
Vl = -54.5;
gNa = 120;
gk = 36;
gl = 0.3;
```

Resuelvo el sistema de ecuaciones diferenciales para diferentes valores de la corriente externa, variando en un rango de 2mA a 50mA.

```
In[21]:= freqData = {};
plots = {};
Table[res = NDSolve[{V'[t] ==
      1/Cap (Current - gNa m[t]^3 h[t] (V[t] - VNa) - gk n[t]^4 (V[t] - Vk) - gl (V[t] - Vl)),
     m'[t] == (m∞[V[t]] - m[t]) / taum[V[t]],
     h'[t] == (h∞[V[t]] - h[t]) / tauh[V[t]],
     n'[t] == (n∞[V[t]] - n[t]) / taun[V[t]],
     V[0] == -80, n[0] == 0, h[0] == 0, m[0] == 0}, {V, m, h, n}, {t, 0, 200}];
  results = res[[1, 1]];
  Vt = V /. results;
  (*data = Abs[Fourier[Table[Vt[t],{t,20,200,0.01}]]];*)
  peaks = FindPeaks[Table[Vt[t], {t, 20, 200, 0.1}], 10, 0, 0];
  If[Length[peaks] > 0,
   frequency = 1/ (Mean[Differences[First /@ peaks]] * 0.1),
   frequency = 0;
  ];
  AppendTo[freqData, {Current, frequency}];
  legend =
   "Corriente = " <> ToString[Current] <> "mA, Frequ = " <> ToString[frequency];
  If[Mod[(Current - 2) / 0.1, 10] == 0,
   AppendTo[plots, Plot[Vt[t], {t, 0, 100}, PlotRange → All,
     Epilog → {Red, PointSize[0.02], Point[Map[{#[[1]] * 0.1 + 20, #[[2]]} &, peaks]]},
     ImageSize → Medium, PlotLegends → Placed[legend, Top]]]]
  (*p2 = ListLinePlot[data,PlotRange→{{2,20},{0,1000}}];*)
  , {Current, 2, 50, 0.1}];
```

```
ListLinePlot[freqData, ImageSize → Large,
  → True, GridLines → Automatic, Axes → True,
 FrameLabel → {"Corriente[mA]", "Frecuencia [kHz]"}]
```



Se observa que alrededor de 7.6 mA comienzan a aparecer spikes con una determinada frecuencia y a medida que se continua aumentando la corriente externa dicha frecuencia aumenta.

Ejercicio 4

Fijo la corriente a cero

In[24]:=
```
Current = 0;
```

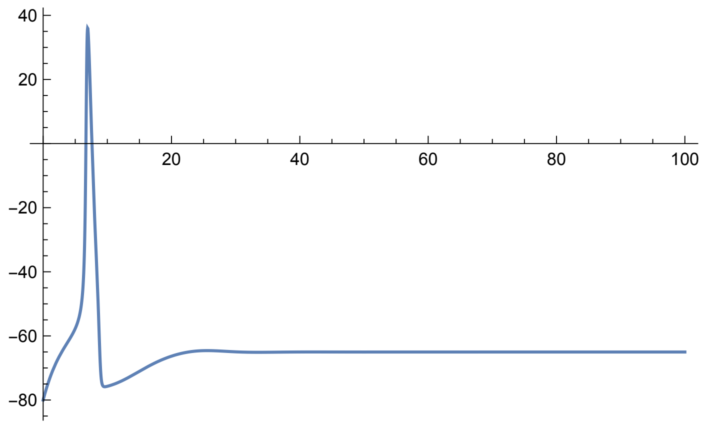
Resuelvo el sistema de ecuaciones

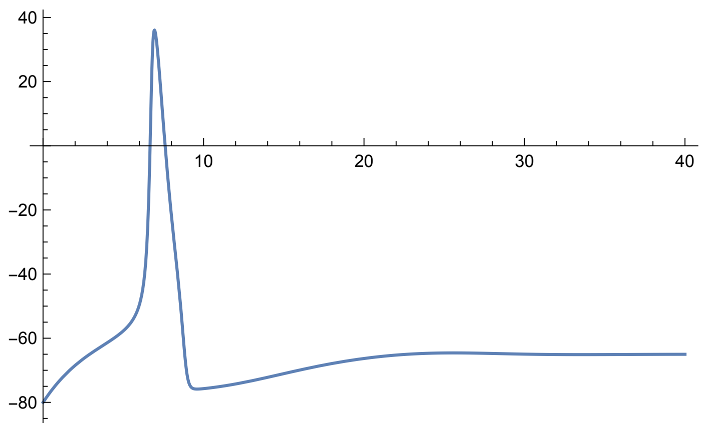
In[25]:=
```
res = NDSolve[{V'[t] ==
      1 / Cap (Current - g_Na m[t]^3 h[t] (V[t] - V_Na) - g_k n[t]^4 (V[t] - V_k) - g_l (V[t] - V_l)),
    m'[t] == (m_∞[V[t]] - m[t]) / tau_m[V[t]],
    h'[t] == (h_∞[V[t]] - h[t]) / tau_h[V[t]],
    n'[t] == (n_∞[V[t]] - n[t]) / tau_n[V[t]],
    V[0] == -80, n[0] == 0, h[0] == 0, m[0] == 0}, {V, m, h, n}, {t, 0, 200}];
VRule = res[[1, 1]];
mRule = res[[1, 2]];
```

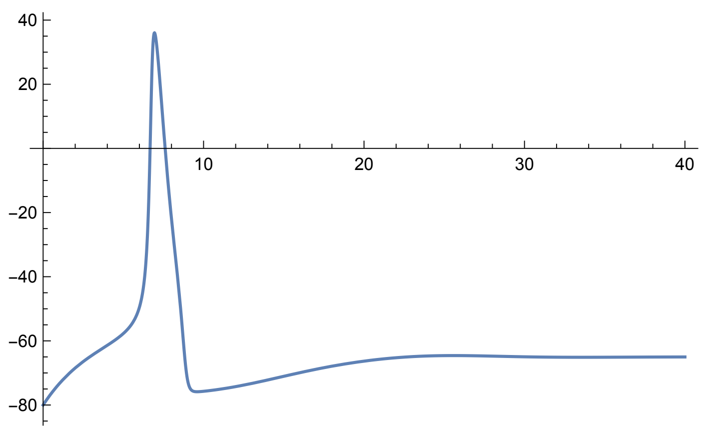
General: Input expression InterpolatingFunction[{{0., 200.}}, <>] contains insufficient information to interpret the result.

In[28]:= `Plot[Evaluate[V[s] /. VRule], {s, 0, 100}, PlotRange → All]`

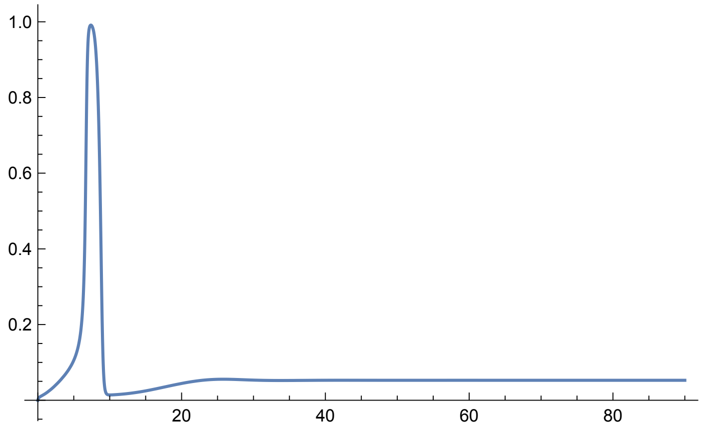
Out[28]=







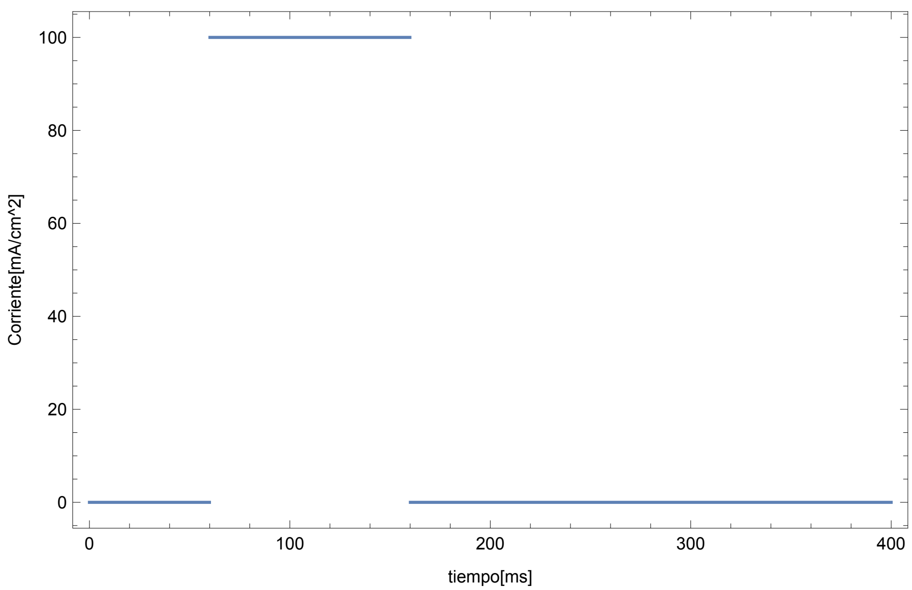
```
Plot[Evaluate[m[t] /. mRule], {t, 0, 90}, PlotRange → All]
```



El sistema alcanza un punto fijo luego de 60ms (un poco antes en realidad, pero por seguridad tomo 60ms).

```
CurrentPulse[t_] := If[t < 60, 0, If[t > (60 + 100), 0, 100]]
```

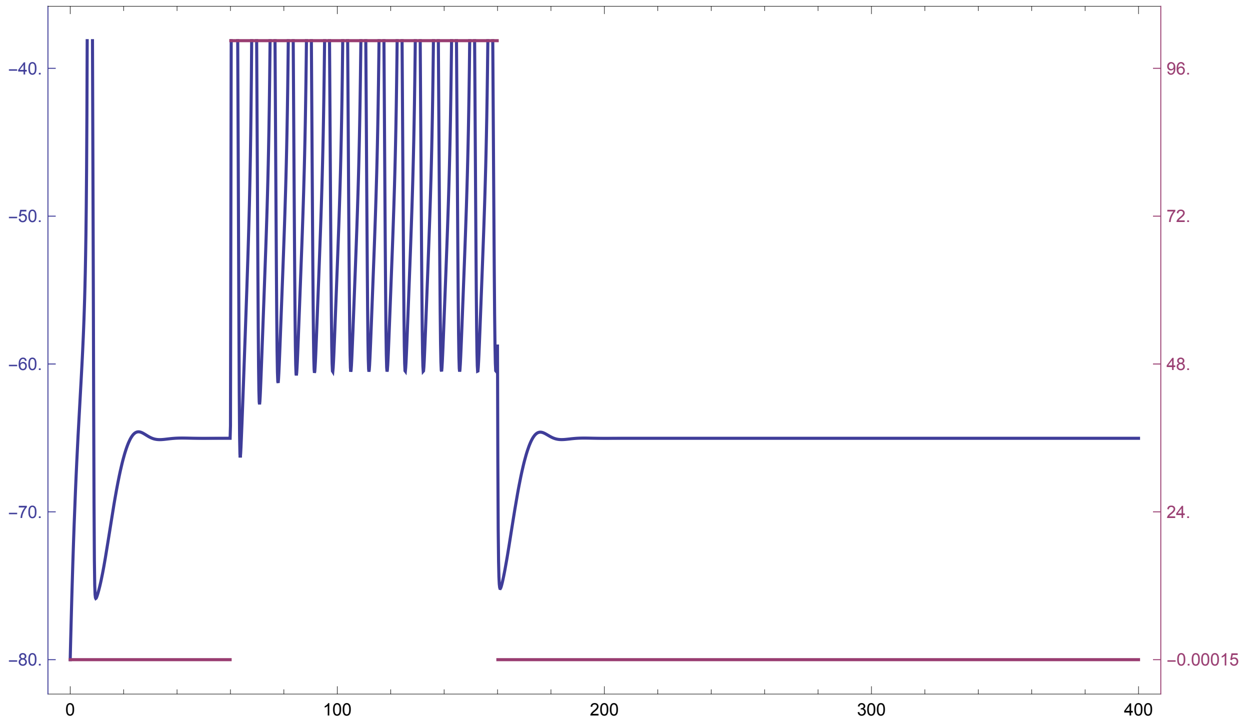
```
Plot[CurrentPulse[t], {t, 0, 400}, Frame → True,
 FrameLabel → {"tiempo[ms]", "Corriente[mA/cm^2]"}]
```



Ahora resuelvo el sitema con este pulso de corriente

```
res = NDSolve[{V'[t] == 1 / Cap
      (CurrentPulse[t] - g_Na m[t]^3 h[t] (V[t] - V_Na) - g_k n[t]^4 (V[t] - V_k) - g_l (V[t] - V_l)),
    m'[t] == (m∞[V[t]] - m[t]) / tau_m[V[t]],
    h'[t] == (h∞[V[t]] - h[t]) / tau_h[V[t]],
    n'[t] == (n∞[V[t]] - n[t]) / tau_n[V[t]],
    V[0] == -80, n[0] == 0, h[0] == 0, m[0] == 0}, {V, m, h, n}, {t, 0, 400}];
```

```
TwoAxisPlot[{Evaluate[V[t] /. res], Evaluate[CurrentPulse[t]]}, {t, 0, 400}]
```



En el grafico de arriba se muestra como evoluciona el sistema luego de activar una corriente de 100 mA/cm^2 por 100 ms.
Luego de que se apaga la corriente el sistema deja de emitir pulsos y regresa al punto fijo.

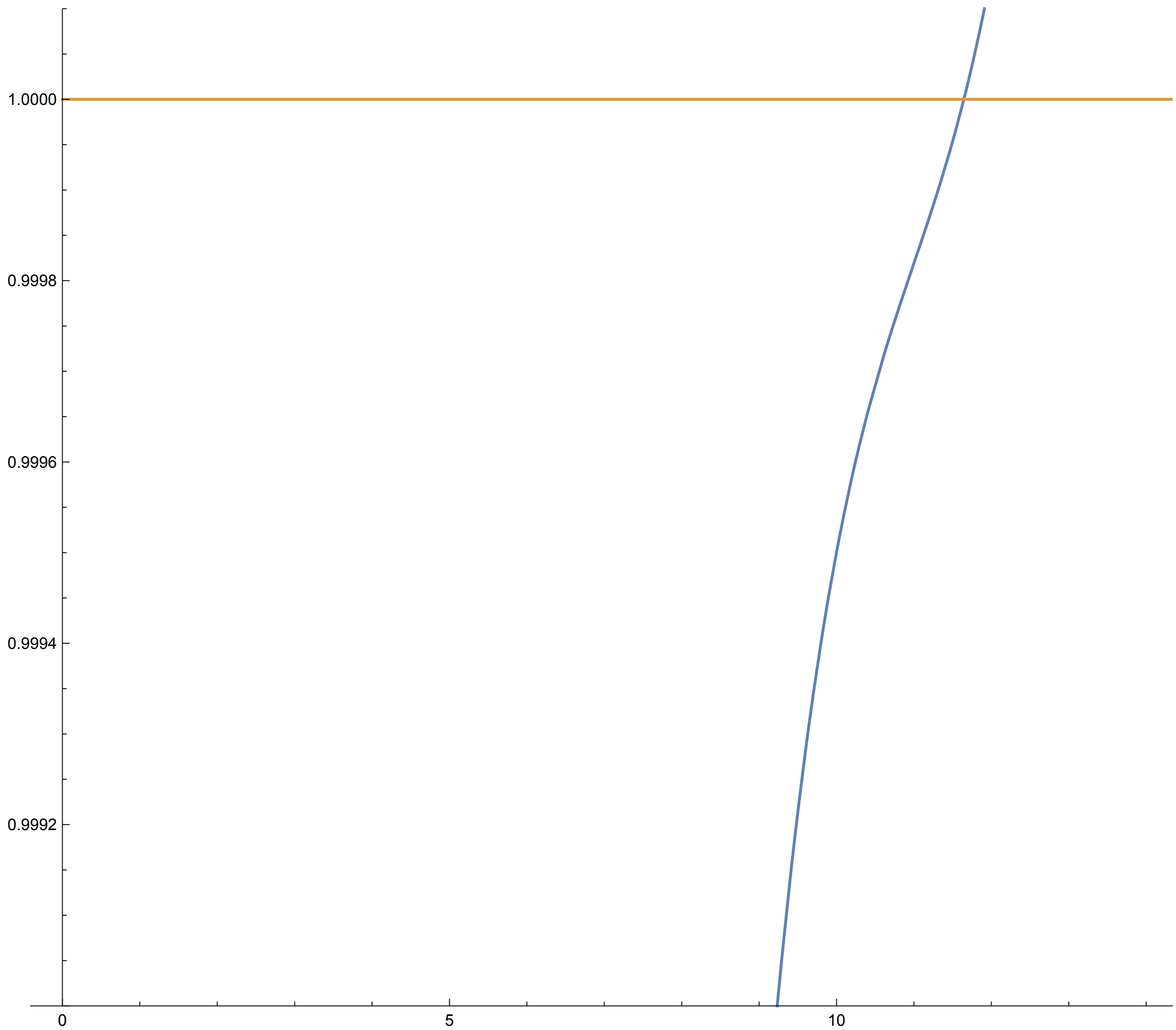
Ejercicio 5

In[29]:=
```
Current = 1;
tau = 1;
tau_a = 1;
```

In[32]:=
```
res = NDSolve[{tau V'[t] == Current - A[t] - V[t],
    tau_a A'[t] == -A[t], V[0] == 0, A[0] == 1}, {V, A}, {t, 0, 10}];
```

In[33]:=
```
Plot[{Evaluate[V[t] /. res], 1}, {t, 0, 20}, PlotRange → {0.999, 1.0001}]
```

Out[33]=



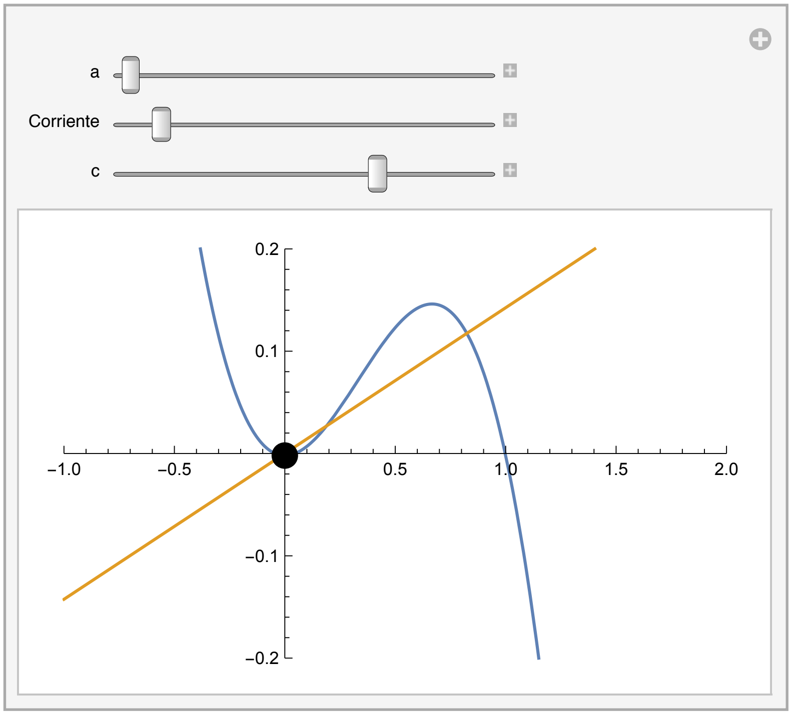
Ejercicio 6

Calculo las nullclinas, obtengo dos sistemas de ecuaciones no lineales,

In[34]:=

```
Manipulate[
 Plot[{x (a - x) (x - 1) + Corriente, c x}, {x, -1, 2}, PlotRange → {{-1, 2}, {-0.2, 0.2}},
  Epilog → {PointSize[0.04], Point[{a, Corriente}]}],
 {a, 0, 1}, {Corriente, -0.1, 1}, {c, 0, 0.2}]
```

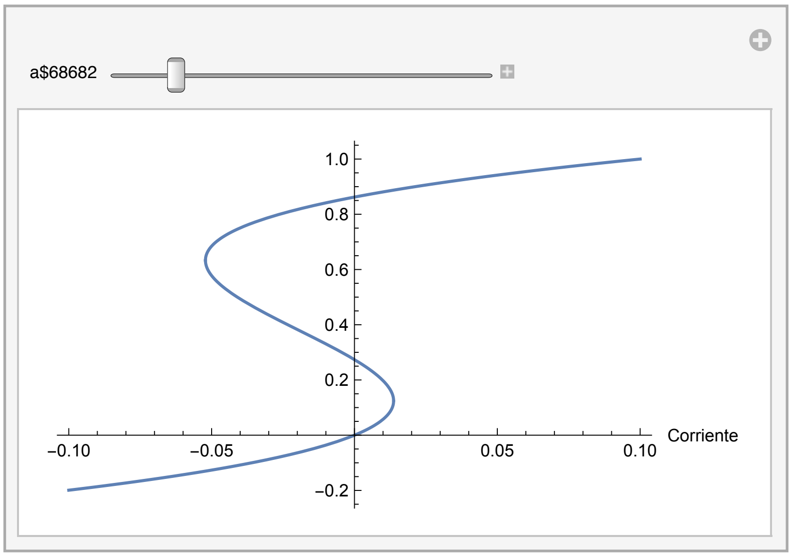
Out[34]=



En la figura de arriba aparecen graficadas la nulclina amarilla, la cual es lineal y la nulclina azul la cual es cubica, se puede ver como al variar la corriente aparecen diferentes puntos fijos.
Fijando a = 0.402 y c= $\frac{b}{\gamma}$=1 se obienen los siguientes puntos fijos en funcion de la corriente

In[35]:=
```
Module[{a, c = 0.1},
 Manipulate[
  f[x_] := x (a - x) (x - 1);
  Plot[x /. Solve[f[x] + Corriente == c x, x],
   {Corriente, -0.1, 0.1}, AxesLabel → Automatic]
  ,
  {a,
   0,
   1}]]
```

Out[35]=



Si el parametro $c = \frac{b}{\gamma}$ es muy grande la curva se hace mas plana y solo aparece un punto fijo todo el tiempo.

Si definimos las funciones F[V,w] y G[V,w] como

```
f[V_] := V (a - V) (V - 2);
F[V_, w_] := f[V] + currnt - w
G[V_, w_] := -γ w + b V
```

Podemos escribir la matriz jacobiana como

In[42]:=
```
jacMatrix[V_, w_] := {{2 a (-1 + V) + (4 - 3 V) V, -1}, {b, -γ}};
jacMatrix // MatrixForm
```

Out[43]//MatrixForm=
```
jacMatrix
```

Usando esta matriz podemos graficar el campo vectorial como

In[44]:=
```
vectorField[V_, w_] := {(2 a (-1 + V) + (4 - 3 V) V) * V, b V - γ w}
```

In[45]:=
```
Manipulate[
  vectorField[V_, w_] := {(2 a (-1 + V) + (4 - 3 V) V) * V, b V - γ w};
  VectorPlot[vectorField[x, y], {x, -2, 2}, {y, -2, 2}],
  {a, 0, 1}, {b, 0, 1}, {γ, 100, 200}]
```

Out[45]=