

# Aplicaciones en electroquímica utilizando software de código abierto (*open source*)

Guillermo A. Flores Martínez\*

## Abstract

En el presente artículo se evalúan diferentes alternativas de uso de software de código abierto en el campo de la electroquímica.

**Palabras Clave:** Python, Simulación, Software, Química Computacional

## Introducción

Un software o programa es de código abierto cuando permite tener acceso a su código de programación, permitiendo al usuario modificar, alterar y hacer con él lo que desee.

## Aplicaciones en electroquímica

Entre las aplicaciones de código abierto que se han desarrollado y que cuentan con elementos relacionados con la electroquímica se tienen los programas:

ASE (Atomic Simulation Environment) [1] y Pymatgen (Python Materials Genomics) [2], ambos basados en Python como lenguaje de programación.

Python es un lenguaje de programación multipropósito, orientado en objetos y cuenta también con soporte para paquetes y módulos (programas). Es importante destacar que la ventaja principal que tienen ambos programas es que, al estar escritos en este lenguaje, permite que puedan ser ejecutados bajo cualquier sistema operativo, evitando así restricciones de compatibilidad.

## Ejemplos

ASE, es un programa que permite efectuar cálculos, visualizar y analizar simulaciones a escalas atómicas y electrónicas, también permite realizar el cálculo de propiedades termoquímicas, elaborar diagramas de equilibrio y Pourbaix [3], siendo estos últimos los que se ejemplificarán en este artículo.

Pymatgen, es un programa orientado al análisis de propiedades de materiales, tales como la densidad de estados electrónicos, estructura de bandas, estados de transición de reacciones químicas (por el método Nudged Elastic Band, NEB), así como la elaboración de diagramas de fase y de Pourbaix [4].

Kinetiscope, es un programa que permite la simulación de reacciones químicas, bajo diferentes condiciones y métodos, tales como reacciones fotoquímicas, catalíticas y polimerizaciones entre otras [5]. Además, permite simular experimentos como voltamperometría cíclica y al mismo tiempo poder obtener propiedades del sistema.

Universidad Iberoamericana Estudiante de 6° semestre de Ingeniería Química., \*gfm1618@outlook.com

```
#se importa el programa:
import ase

#se importan los módulos:
from ase.phasediagram import Pourbaix , solvated

"""
Diagrama de Pourbaix Fe
"""
#se crea una lista con los equilibrios de la especie de interés:
refsFe = solvated('Fe')

#se agrega la especie de interés a la lista:
refsFe += [('Fe',0)]

#se crea un intervalo de pH y potencial:
U = np.linspace(-3, 3, 600)
pH = np.linspace(-2, 16, 600)

#se crea el objeto a partir de la especies:
pFe = Pourbaix(refsFe, Fe=1 ,0=1)

#se grafica el digrama al intervalo de pH y potencial:
d, names, text = pFe.diagram(U, pH, plot=True)
```

Figura 1. Código para construir un diagrama de Pourbaix en ASE.

```
#se importa el programa y módulos:
from pymatgen import MPRester
from pymatgen.analysis.pourbaix_diagram import PourbaixPlotter
from pymatgen.analysis.pourbaix_diagram import PourbaixDiagram

"""
Diagrama de Pourbaix Fe
"""

mpr = MPRester('ORINSL5UHfnJ9qEa')

#se crea una lista con las especies de interés:
entries = mpr.get_pourbaix_entries(['Fe'])

#se crea el objeto a partir de las especies:
pFe = PourbaixDiagram(entries)

#se crea la gráfica:
plot = PourbaixPlotter(pFe)

#se grafica:
plot.get_pourbaix_plot().show()
```

Figura 2. Código para construir un diagrama de Pourbaix en Pymatgen.

Como se puede observar en las figuras 1 y 2 se tiene que especificar la especie sobre la cual se quiere conocer el equilibrio, además se pueden especificar los intervalos de E y pH, así como la concentración de las especies en solución. Posteriormente se llama al módulo de gráficos mediante el cual se producen los diagramas.

En la figura 3, se muestra cómo se puede configurar la simulación de una voltamperometría cíclica, por ejemplo, las condiciones de temperatura, el barrido de potencial (E), etc.

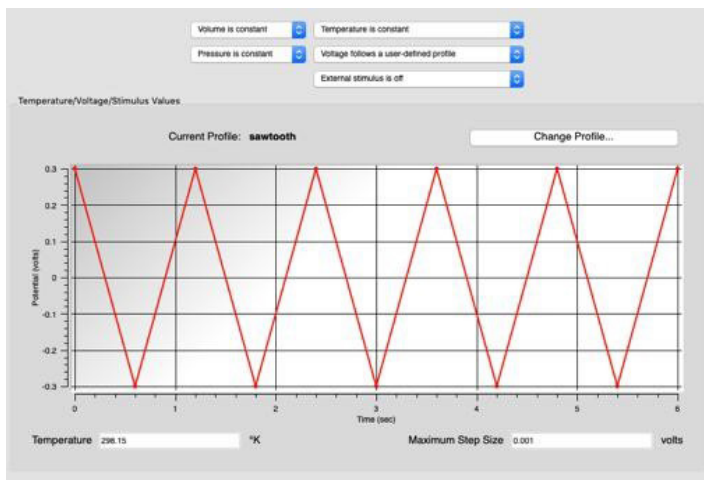


Figura 3. Simulación de una voltamperometría cíclica en Kinetoscope.

Como se puede observar en los diagramas obtenidos en las figuras 4 y 5, se cuenta con una diferencia en el número de especies de Fe, siendo el diagrama obtenido por Pymatgen más completo; sin embargo, al observar el que se obtuvo en ASE, se notan los equilibrios con diferentes colores, lo cual permite una rápida distinción y facilita la lectura del mismo.

### Conclusiones

Las aplicaciones en electroquímica revisadas demuestran tener gran utilidad tanto para el aprendizaje como para su implementación en experimentos e investigaciones. Además, al ser software de código abierto, permite que cualquiera pueda tener acceso sin que se esté sujeto a la adquisición de una licencia.

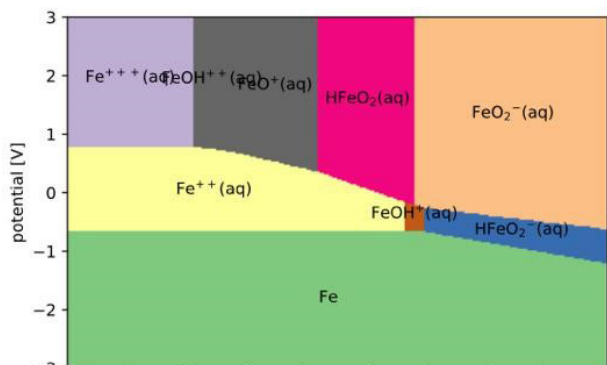


Figura 5. Diagrama de Pourbaix de Fe obtenido en Pymatgen.

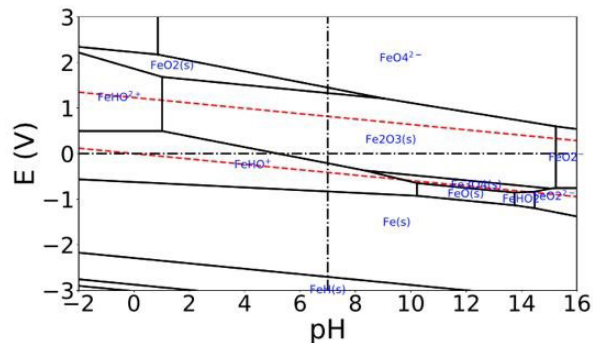


Figura 4. Diagrama de Pourbaix de Fe obtenido en ASE.

### Referencias

- Hynninen, T.; Himanen, L.; Parkkinen, V.; Musso, T.; Corander, J.; Foster, A. S. An object oriented python interface for atomistic simulations, *Comput. Phys. Commun.*, **2016**, *198*, 230–237. <https://doi.org/10.1016/j.cpc.2015.09.010>
- Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G.; Persson, K. A. Commentary: The materials project: A materials genome approach to accelerating materials innovation, *APL Mater.* **2013**, *1*, 011002. <https://aip.scitation.org/doi/10.1063/1.4812323>
- Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Duřak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Jensen, P. B.; Kermode, J.; Kitchin, J. R.; Kolsbjerg, E. L.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Maronsson, J. B.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The atomic simulation environment a python library for working with atoms, *J. Phys.-Condens Mat.*, **2017**, *29*, 273002. <http://stacks.iop.org/0953-8984/29/i=27/a=273002>
- Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, *Comp. Mater. Sci.*, **2013**, *68*, 314–319. <https://doi.org/10.1016/j.commatsci.2012.10.028>
- Ranke, J.; Wöltjen, J.; Meinecke, S. Comparison of software tools for kinetic evaluation of chemical degradation data, *Environ. Sci. Eu.* **2018**, *30*, 1, 17; 17–17. <https://doi.org/10.1186/s12302-018-0145-1>