[Nombre del proyecto]

Estándar de Implementación

Versión 1.2

Historia de revisiones

Fecha	Versión	Descripción	Autor
21/08/2013	1.0	Creación del documento	Alejandro Cardone, Viterbo García
24/08/2013	1.1	Ajustes al documento	Agustín Azzinnari, Miguel Merlino
25/08/2013	1.2	Modificación de acuerdo a los estándares	Gonzalo Antúnez

Contenido

1. Convenciones Generales

1.1. Legibilidad

- Los nombres de los identificadores y parámetros deben de favorecer la legibilidad sobre la brevedad, por ejemplo, es preferible usar datos_estudiante_avanzado sobre datos_est_av; salvo que sean reconocidas internacionalmente (por ejemplo xml, ok).
- Usar acrónimos sólo cuando sean necesarios.

1.2. Idioma

- La nomenclatura (nombre de variables, clases, métodos) debe estar en idioma inglés.
- Los comentarios deben estar en idioma español.

1.3. Comentarios

- Realizar comentarios donde el código no sea lo suficientemente intuitivo.
- No comentar innecesariamente ni de forma inadecuada. Por ejemplo, este comentario es innecesario:
  x = x + 1 # Sumo 1 a x
  Éste es útil:
  x = x + 1 # La vaca tuvo una nueva cría.

2. Convenciones Específicas

2.1. Variables

- Deben empezar con un letra minúscula o un guión bajo (_).
- Deben estar formadas por letras minúsculas, números y/o guiones bajos
- Usar snake_case [2].

2.2. Constantes

- Usar SCREAMING_SNAKE_CASE (snake_case utilizando solo mayúsculas).
- Utilizar nombres descriptivos que reflejen el significado y no el tipo.
- Los comentarios sobre una constante se deben hacer en la misma línea.

2.3. Clases

- No definir más de una clase en un mismo archivo.
- Los nombres de las clases deben utilizar UpperCamelCase [1], y deben ser sustantivos en singular. Correcto: Oveja, AnimalVerde. Incorrecto: oveja, Animales, Animalvacuno.
- Los atributos de una clase deben estar definidos antes de los métodos.
- Los atributos de una clase no deben ser públicos.
- La privacidad de los métodos y atributos (public, private o protected) debe definirse en una línea anterior. Por ejemplo:

```
class Clase
        private
@atributo1
```

```
        @atributo2
        ...

        public
        def get_atributo1
            ...
        end
        ...
    end
```

2.4. Metodos

2.4.1. Nomenclatura

- Los nombres de los métodos deben denotarse utilizando snake_case [2]. Por ejemplo, son correctos: obtener_stock, registrar_cria; son incorrectos: obtenerStock, RegistarCria.
- Para los predicados (métodos que devuelven booleanos), utilizar una frase afirmativa con un signo de iterrogación al final. Por ejemplo: es_vacio?, que devuelva true si es vacío. Opcionalmente, también puede prefijar las propiedades booleanas con *es*, *tiene* o *puede*.

2.4.2. Llamados

- No utilizar paréntesis en los métodos que no reciben parámetros. Por ejemplo, utilizar Vaca.vacunar en lugar de Vaca.vacunar().

2.5. Indentación

- Escribir una sentencia por línea.
- Utilizar 2 espacios para tabular.
- Todo bloque debe tener un tabulador más de identación que el bloque que lo contiene.
- Las sentencias y bloques dentro de un bloque deben estar identadas con un tabulador más que el bloque que las contiene.
- Los comentarios sobre un bloque deben estar en la/s línea/s anterior/es al bloque sin dejar líneas en blanco entre la última línea del comentario y el comienzo del bloque y dejando una línea en blanco antes del comentario.
- Los inicios y cierres de bloques deben estar en la misma columna.

```
class ...
  def ...

  end
end
```

3. Otros

- Para los nombres de módulos usar UpperCamelCase [1].

- Usar espacios alrededor de los operadores, después de las comas, dos puntos y punto y coma, alrededor de { y } antes.
- No usar espacios después de (,[y antes de],).
- Utilizar una línea vacía antes del valor de retorno de un método.
- Utilizar líneas vacías para dividir un procedimiento largo en párrafos lógicos.
- En la medida de lo posible, no sobrecargar un procedimiento o método con demasiada funcionalidad. Para esto, dividir lógicamente el procedimiento en subprocedimientos que favorezcan el entendimiento y comprensión del comportamiento del mismo.
- Mantener las líneas de menos de 80 caracteres.
- No utilice métodos que tengan los mismos nombres que los métodos Get.
- Evitar espacios en blanco.

4. Referencias

[1] http://es.wikipedia.org/wiki/CamelCase

[2] http://en.wikipedia.org/wiki/Snake_case