



Escuela Politécnica Nacional

Facultad de Ingeniería de Sistemas

Programación I

Estándares de desarrollo de software para la programación de un Sistema de Check In/Out para un Hotel

Equipo 1

Integrantes:

- Jahir Rocha
- Sara Rosero
- Najely Salas
- Kevin Tacuri

Contenido

1. Propósito	3
2. Destinatario	3
3. Plataforma C++	3
3.1 Buenas Prácticas	3
4. Referencias	4

1. Propósito

El objetivo de este documento es explicar las reglas fundamentales que debe seguir el equipo de desarrollo de software al codificar una aplicación para que sea fácil de entender, mantener y mejorar.

2. Destinatario

Integrantes del grupo número 5 que desarrollan un sistema con buenas practicas de programación en Arduino y C++.

3. Plataforma C++

3.1 Buenas Prácticas

Idioma

- Se utilizará el idioma español para nombrar constantes, variables, procedimientos, funciones, tablas y atributos. Excepto para las funciones y procedimientos que sean de librerías preestablecidas.

Estructura de código

- Para mejorar la comprensión del código, las variables, funciones y procedimientos deben nombrarse utilizando la convención de Camelcase (la primera letra en minúscula y las palabras siguientes en mayúsculas sin espaciado) en c++ y Arduino.
- Usar prj antes de indicar el nombre de una función o procedimiento: prjEliminarUsuarioCheckOut(). Y estos deberán ser correctamente documentados usando `/**/` en este se incluirá que tipo de variable usa y que proceso realiza esta función o procedimiento.
- Las constantes y variables serán etiquetadas con nombres significativos: "estadoCheck" indicará de manera adecuada si ha completado el check-in.
- En las bibliotecas, también se debe incluir prj antes de etiquetarlas: prjColor.h. Y usar las necesarias para no sobrecargar el código.
- Las secciones de código se separarán claramente utilizando comentarios y líneas en blanco. Además, se utilizará sangría consistente y espaciado para mejorar la legibilidad.
- Si necesita una explicación adicional de una variable o función en el código, use los comentarios.
- Si hay constantes repetidas en el código, se definen al principio del código: `#DEFINE DELAY 100000`.

Backup

- Se deben generar backups lógicos full export periódicos de cada proceso de producción en su respectivo filesystem bajo el directorio: <https://github.com/Jahir13/prjGrupo05>

4. Referencias

- [1] “C++ Core Guidelines,” [isocpp.github.io](https://isocpp.github.io/CppCoreGuidelines/). <https://isocpp.github.io/CppCoreGuidelines/> (accessed Jun. 29, 2023).
- [2] H. Sutter and Andrei Alexandrescu, C++ Coding Standards. Pearson Education, 2004.
- [3] “C++ 2019 - The state of Developer Ecosystem in 2019 Infographic,” JetBrains. <https://www.jetbrains.com/lp/devecosystem-2019/cpp/> (accessed Jun. 29, 2023).
- [4] S. Meyers, Effective C++. Addison-Wesley Professional, 1998.