



[Escuela de código]

Apuntes de clase:

Guía de proyecto integrador

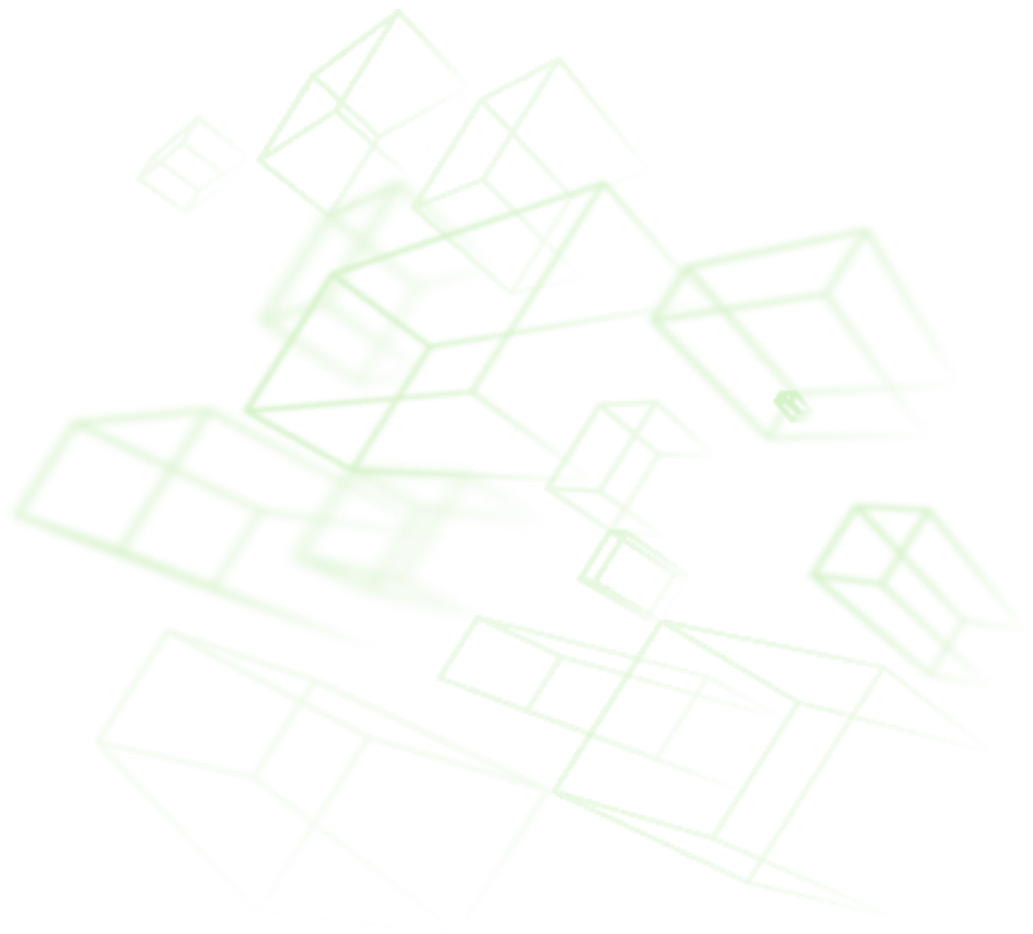


Tabla de contenido

<i>Proyecto integrador: Objetivos</i>	2
<i>Validación</i>	2
<i>Presentación</i>	5
<i>Trabajamos los bloques individuales</i>	6
<i>Prueba de concepto</i>	6
<i>Certificación</i>	6
<i>Vías de consulta y avance</i>	7

Proyecto integrador: Objetivos

El objetivo del proyecto integrador es poner en práctica los conocimientos obtenidos a lo largo de la cursada, volcándolos en un proyecto propuesto por los alumnos, que los docentes evaluarán para poder validar y así certificar que han adquirido los conocimientos de este curso.

Los alumnos deben proponer un proyecto y resolver los obstáculos que se les presenten en el camino, de manera de forjar una forma de trabajo que les permitan desenvolverse solos, siendo lo más parecido a la vida real. Los docentes acudirán a su ayuda solo en el contexto en el que el alumno no pueda avanzar a pesar de contar con el material brindado en el curso o el disponible en la web.

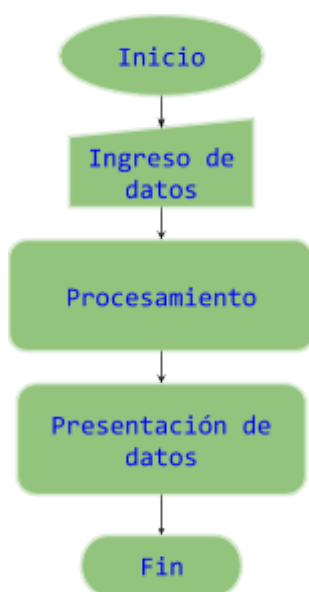
Validación

Para comenzar, deben definir un proyecto, una aplicación, que como mínimo involucre los módulos vistos en el curso, por lo que no se requiere que el proyecto deba ser más complejo que un trabajo práctico visto en clase.

Se brindan algunos ejemplos en el foro de proyecto, para tener como referencia.

Aconsejamos:

- Buscar alguna aplicación que pueda resolver algún problema o proceso de su vida cotidiana, de manera que pueda ser útil al ser terminada.
- Pensar a futuro, es decir, que las funcionalidades de su aplicación puedan ser ampliadas con los siguientes cursos.
- Que la aplicación tenga alguna finalidad comercial. Ésto tracciona su evolución.



Una vez definido el “que vamos a hacer”, definamos el “cómo vamos a hacer”, para ello, recomendamos estructurar su programa, generando un diagrama en bloques, que puede ir desde lo más sencillo, a lo más complejo.

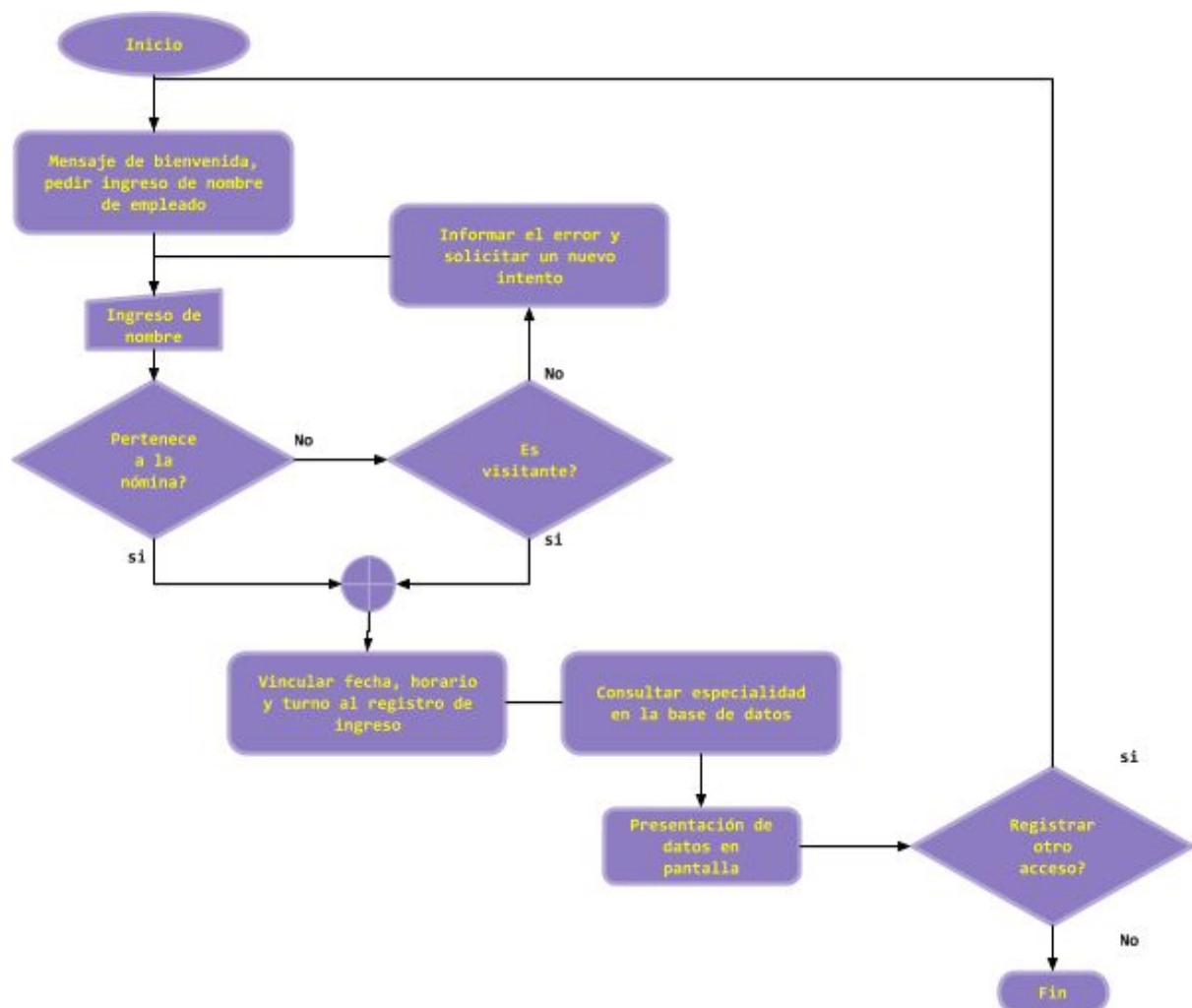
Para elaborar el diagrama de flujo, pueden utilizar cualquier herramienta que deseen, hasta una hoja de papel escrita a mano, pero después deben presentarlo como parte del proyecto. Una herramienta libre es: <https://app.diagrams.net/> Pero pueden utilizar cualquier otra.

La idea de estructurar el proyecto es dividirlo en pequeños bloques, en donde cada uno se encarga de una tarea específica. De esta manera lo que puede verse como un gran problema a resolver, se transforma en pequeños problemas a solucionar.

Esta forma de realizar el trabajo nos permite estimar la factibilidad del proyecto, el tiempo de ejecución y determinar los procesos críticos que pueden llevar la mayor parte del tiempo.

Por ejemplo, nos dan el requerimiento de guardar en un archivo el control de acceso del personal en una fábrica, indicando el nombre, el turno, el horario y el sector.

Planteamos el diagrama de flujo:



A medida que planteamos el diagrama de flujo, se va ir evidenciando las herramientas que vamos a necesitar para cumplir con el proyecto y con cada bloque en particular, por ejemplo, para este caso, con las herramientas más básicas podemos:

- **Mensaje de bienvenida:** Imprimir por consola el mensaje y solicitar el ingreso del nombre del empleado, notificando cuales son los caracteres válidos e inválidos.
- **Ingreso de nombre:** Ingreso por teclado y asignación a una variable para ser analizada
- **Pertenece a la nómina?:** Comparamos la variable ingresada con una base previamente guardada en un archivo (no podemos utilizar listas porque su modificación se mantiene en RAM, con lo cual, no se mantiene una vez cerrada la aplicación)

- **Es visitante?** Caso en que no se encuentre consultar si se ingresan sus datos como visitante para el registro, estos datos deben guardarse en un archivo.
- **Vincular datos de fecha, hora y turno con el usuario:** Se debe consultar el horario y la fecha en el sistema y junto con el nombre del empleado guardarlo en una lista dinámica.
- **Consultar la especialidad del usuario:** Buscar en el archivo de base de datos, la especialidad del usuario y asignarlo a la lista dinámica con el nombre de usuario que se está ingresando. En caso de ser visitante, usar este dato como especialidad.
- **Presentación de los datos en pantalla:** Imprimimos por consola los datos de la lista que tenemos cargada y la guardamos en el archivo de base de datos.
- **Registrar otro Acceso?:** consultamos por consola si se quiere ingresar otro usuario y esperamos la respuesta por teclado [Y/n]
 - En caso de no, enviamos un mensaje de agradecimiento y cerramos la aplicación.

Este pequeño ejemplo nos permite obtener una visión tanto global como en detalle de la arquitectura del software, obteniendo así los puntos críticos para el diseño, los cuales pueden ser:

- Que formato tiene que tener el archivo de base de datos?
- Se va a trabajar con un solo archivo o con varios a la vez?
- Como obtengo el horario del sistema?

Así mismo, podemos evidenciar las características que se pueden sumar en un futuro, por ejemplo:

- Una contraseña para poder abrir la aplicación, sumando seguridad al proceso.
- Una interfaz gráfica para el ingreso o selección del usuario a ingresar.
- Gráficas y alertas del horario de acceso de el o los usuarios.

Esto es solo un ejemplo de cómo realizar un proyecto, aunque para nada significa que es la única manera, existen varios métodos, pero solo brindaremos éste como puntapié inicial.

Presentación

La idea de la presentación es tener una manera ordenada y prolija en la que todos aquellos que realicen un proyecto puedan exponerlo. Ésto es de gran ayuda para el día de mañana, si desean tener un portfolio de trabajos, puedan utilizarlo para sus búsquedas laborales.

El proyecto debe presentarse en github, en la cuenta de cada alumno, y luego enviar el link del repositorio por medio del foro del campus.

Para la presentación es necesario indicar:

- **Nombre del proyecto:** Puede ser simplemente una breve descripción de su uso o un nombre personalizado.
- **Descripción:** Descripción detallada del proyecto (que es y para qué sirve), en qué lenguaje está realizado, como funciona, parámetros de entrada/salida, etc;
- **Diagrama de flujo.**
- **Ejemplos.**
- **Fecha de última actualización.**

Esto debe hacerse en el archivo **README.md** y de preferencia les aconsejamos utilizar imagenes que refieran a la descripción del proyecto, de manera de hacerlo más "atractivo" para aquellos interesados en su perfil.

Importante: *No es obligatorio (pero si aconsejable) que el archivo README.md tenga un formato específico, o atractivo, ni tampoco que tenga imágenes, simplemente basta con que especifique los ítems listados anteriormente.*

Como ejemplo, pueden ver los siguientes repositorios:

- <https://github.com/InoveProyectos/Buscador-Alquileres-Python>
- <https://github.com/TexasInstruments/tensorflow>
- <https://github.com/RedHatOfficial/ocp4-helpernode>

Para la edición del archivo README.md pueden utilizar el instructivo brindado en el campus, u otra fuente de su preferencia.

Trabajamos los bloques individuales

La idea de dividir el programa en bloques, es justamente poder resolverlos de manera individual. Para ello tenemos que tratar cada bloque como una pequeña (o no tanto) fracción del código que resuelva solo la tarea encomendada, de la manera más genérica posible, de manera que este bloque pueda ser reutilizado. Para lograr esto, debe tenerse muy en claro la arquitectura del programa.

Los datos de entrada de cada bloque deben ser generados a mano o por medio de muestras que obtengamos para su aplicación, los datos de salida deben ser aptos para que los reciba el siguiente proceso.

Prueba de concepto

En esta etapa, se busca realizar una prueba que confirme que lo planteado es posible implementar en la vida real, es decir, que nuestro diagrama de flujo pueda "salir del papel" para volcarlo en algo real. Esta es una manera de evidenciar los imprevistos a la hora de ejecutar nuestra aplicación, por ejemplo, en el caso de análisis antes planteado:

Puede ser que el programa deba ser ejecutado en varios sistemas operativos a la vez, algo que no estaba contemplado en los requerimientos, pero sucede que nuestro cliente tiene varios sistemas operativos corriendo en distintos ingresos a su planta...

Este es un imprevisto como tantos otros que pueden aparecer y que se evidencian a la hora de volcar la aplicación a un entorno real.

Esto muchas veces requiere de cambios en la arquitectura planteada inicialmente para poder abordar a su solución, de ser así, debe modificarse el diagrama de flujo y el programa, pero vamos a tener una aplicación cada vez más cercana a la final.

Certificación

Los profesores evaluarán el estado del proyecto. Muchas veces el proyecto excede a lo visto en el curso, y es por ello que no se requiere que el mismo esté terminado. Sin embargo, es imperativo que en el mismo:

- Se vuelquen los contenidos vistos en clase.
- Se pueda evidenciar el funcionamiento de al menos las partes que utilizan las herramientas vistas en clase.
- Se entregue el enlace al repositorio con la descripción del proyecto.

Luego del análisis de los docentes, si el alumno cumple con estos requisitos, se entenderá que alcanzó los conocimientos sobre el contenido de la cursada y le otorgamos la certificación emitida por inove.

Vías de consulta y avance

Siempre el medio de comunicación entre docentes y alumnos es por medio del campus de inove. Propiamente en este caso el foro de la unidad de proyecto. Cada alumno debe plantear sus inquietudes, así como avances para que el profesor pueda ir evaluando en el transcurso del proyecto y para enriquecer el conocimiento del grupo.

El avance depende de los alumnos, los docentes solo brindarán su colaboración para evacuar alguna duda relacionada con la arquitectura del proyecto, y para la validación, pero no para resolver dudas sobre cuestiones vistas en clase (para esto puede consultarse la documentación en el campus u otra fuente externa). Esto es para fomentar la autonomía de cada alumno y que puedan desenvolverse solos para resolver los problemas que plantea cada proyecto, acercandolos a una situación más real, en donde el rol del cliente en este caso, lo cumplen los docentes.

Los docentes informarán a los alumnos cuando hayan alcanzado las metas de la certificación, que posteriormente les serán brindados por el departamento de alumnos.