Cómo crear una aplicación

¿Qué es una aplicación?
¿Qué es crear una aplicación?
¿Cómo se crea una aplicación?

¿Qué es una aplicación?

Una aplicación o programa es la ejecución de instrucciones lógicas de uso específico expuestas a la interacción mediante una interfaz utilizable por un ser humano.

¿Qué es crear una aplicación?

La creación de una aplicación es el proceso de producción de un sistema utilizable uniendo tres elementos:

1. Instrucciones lógicas
2. Ejecución de instrucciones lógicas
3. Exposición a una interfaz interactuable

¿Cómo se crea una aplicación?

Para cada uno de los 3 elementos del proceso de desarrollo o creación de una aplicación (definición de instrucciones lógicas, su ejecución y la exposición a una interfaz interactuable) podemos definir 4 etapas:

1. Definición
2. Prototipado o diseño
3. Investigación
4. Desarrollo

1. Definición de instrucciones lógicas

1.1. Definición
1.2. Prototipado
1.3. Investigación
1.4. Desarrollo

1. ¿Qué son las instrucciones lógicas?

Por ejemplo: Si quisiéramos contar la cantidad de ciclistas con y sin casco que cruzan una intersección, podemos definir las instrucciones:

- Identificar ciclistas que atraviesan un punto
- Categorizarles según llevan casco o no
- Realizar un conteo de cada categoría

Estas instrucciones no están ligadas a ningún tipo de implementación

Son reglas que se cumplen haya o no aplicación. Lo que se espera que la aplicación haga.

Estas reglas existen antes que la aplicación, son las que le dan el sentido.

En general, la aplicación es una respuesta a ¿cómo convertir estas reglas en procesos utilizables y aplicables en la vida humana?

1.1. Definición de instrucciones lógicas: definición

Coloquialmente listamos (preferentemente en orden) los pasos que hubiera que seguir con el mayor detalle posible y un desmenuzado excesivo para llegar a un resultado esperado.

También, definir el resultado esperado.

I. Identificar ciclistas en determinado punto
II. Identificar si el ciclista lleva o no casco
III. Categorizar al ciclista de acuerdo a si lleva o no casco
IV. Contar categoría “sin casco”
V. Contar categoría “con casco”
VI. Visibilizar resultados

Resultado esperado: Dos números que representen la cantidad de ciclistas con casco y la cantidad de ciclistas sin casco que pasen por un punto determinado, respectivamente.

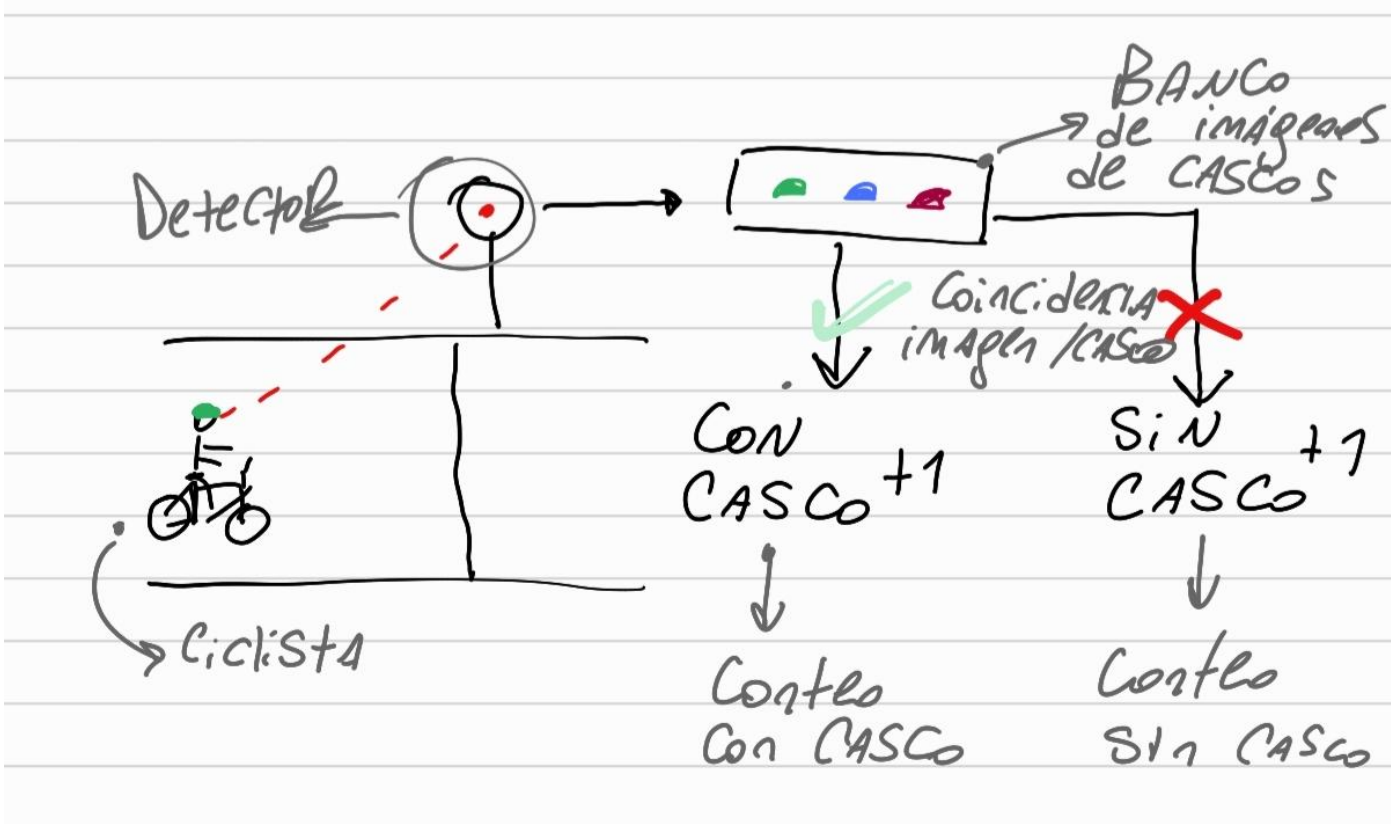
1.2 Definición de instrucciones lógicas: Prototipado

El prototipado de estas reglas sigue siendo un proceso coloquial. Por escrito o en un diagrama, recorremos nuestras reglas lógicas en orden, agregando detalles y haciendo ajustes si fuera necesario.

Por ejemplo, de manera escrita:

Un ciclista con casco cruza la intersección, es detectado como un ciclista con casco en base a la comparación con un banco de imágenes de cascos, se lo categoriza como tal y se le suma 1 al contador de ciclistas con casco.

Dos ciclistas, uno con casco y uno sin, cruzan la intersección, son detectados como dos ciclistas, uno es detectado con casco y el otro sin, se los categoría como tal y se le suma 1 al contador de ciclistas con casco y 1 al contador de ciclistas sin casco.



O en forma de diagrama

1.3 Definición de instrucciones lógicas: investigación

Es hora de saber si nuestra idea conceptual es factible, si el prototipo es realizable. Para esto necesitamos investigar la definición de nuestras entidades, casos de uso similares, posibles inconveniencias, ejemplos, métodos de aplicación, etc.

La investigación (no necesariamente externa) es el paso previo al desarrollo, por lo que en este punto deben cubrirse todas las dudas y se debe tener un panorama claro de cómo las reglas van a existir y funcionar entre sí. Cómo de la nada llegar al resultado esperado.

Es esperado que luego de la investigación haya que reestructurar algunas cosas teniendo que volver sobre el prototipado o hasta sobre la definición. Esto debe ocurrir tantas veces como sea necesario.

1.4 Definición de instrucciones lógicas: desarrollo

Es momento de pensar cómo estas reglas van a cobrar vida en forma de un proceso utilizable

Es en este momento donde decidimos cuál va a ser el ejecutor del proceso:

¿va a ser un ser humano, manualmente contando ciclistas con o sin casco? ¿una planilla que deba ser forzosamente completada por los ciclistas al pasar por la intersección? ¿dos botones físicos en el pavimento por los cuales los ciclistas tengan que pasar si tienen o no casco?

Mejor que sea una app....

2. Ejecución de las reglas lógicas

2.1. Definición
2.2. Prototipado
2.3. Investigación
2.4. Desarrollo

2. Ejecución de reglas lógicas

- Dónde: ¿En una compu? ¿En un teléfono? ¿En una tele? ¿En una página web? ¿En un Arduino? ¿En el STM32?
- Cómo: De qué manera las reglas existen en el dónde. ¿En un celular? Que la cámara del mismo sirva para identificar los ciclista sy en la pantalla se muestre el conteo, ¿En una página web? Mediante una cámara gubernamental capturar las imágenes, detectar ciclistas y mostrar el conteo en el navegador
- Sobre qué: ¿Sobre qué datos van a ser aplicadas nuestras reglas? ¿Sobre imágenes, un video, un reporte humano, respuestas a una encuesta? Los datos a los que tendría sentido aplicarles estas reglas.

Ya conocemos con alto grado de detalle, cuáles y cómo son las reglas lógicas, los límites necesarios para llevar a cabo su implementación, o sea, hacer que se cumplan.

Para esto, vamos a definir un universo conceptual, de dónde, cómo y sobre qué van a poder ser aplicadas.

2.1. Ejecución de reglas lógicas: definición

Convertimos nuestras reglas lógicas en partes, módulos, funciones de única responsabilidad (que hagan una sola tarea) aplicables a un sistema informático.

Respondiendo a las preguntas dónde, cómo y sobre qué, podemos iniciar el prototipado de nuestra aplicación, de la ejecución de las reglas lógicas.

2.2 Ejecución de reglas lógicas: Prototipado

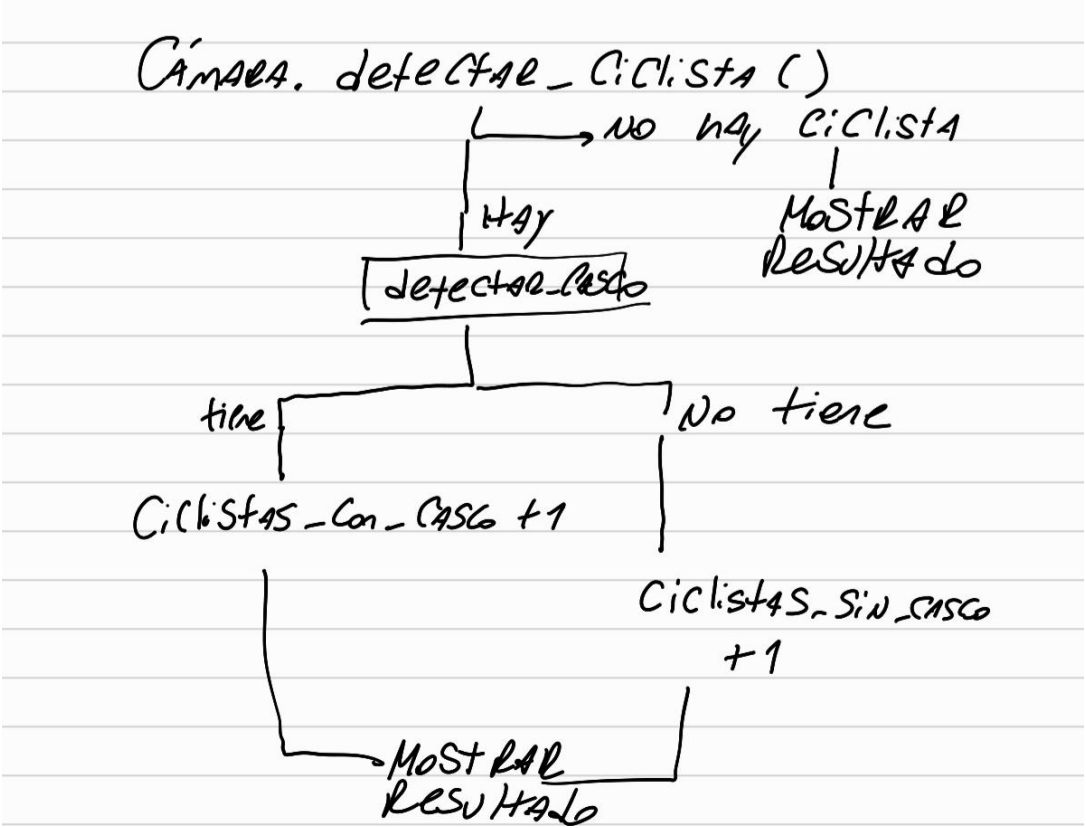
Debemos realizar un prototipo de la lógica programática de la aplicación. Nuevamente, el prototipo puede estar en formato texto o diagrama.

En los prototipos programáticos, suele utilizarse el psuedocódigo: palabras en lenguaje humano estructuradas como si fueran un lenguaje de programación, pero no lo es, suele escribirse con poco grado de detalle.

Por ejemplo, en modo texto:

```
Coleccion<Ciclista> ciclistas;
Ciclista ciclista_detectado;
Camara camara;
numero ciclistas_con_casco;
numero ciclistas_sin_casco;

function main(){
  ciclista = camara.detectar_ciclista();
  si(ciclista){
    agregar(ciclista, ciclistas);
    tiene_casco = camara.detectar_casco(ciclista);
    si(tiene_casco){
      ciclistas_con_casco = ciclistas_con_casco +1;
    }si no{
      ciclistas_sin_casco = ciclistas_sin_casco +1;
    }
  }
  mostrar_resultado();
}
```



O en forma de diagrama

2.3 Ejecución de reglas lógicas: investigación

Una vez más debemos poner en duda nuestra definición y prototipado, buscando posibles fallas, errores de comprensión y ejemplos de implementaciones existentes.

En este paso también investigamos cómo convertir nuestro prototipo en software funcional, aquí elegiremos librerías y tecnologías que usaremos.

Librerías: Código ya escrito por un programador con el único fin de ser usado por otro programador. Por ejemplo, podríamos importar una librería de detección de ciclistas (en vez de tener que programar esa funcionalidad)

Tecnologías: Qué va a ejecutar nuestro código ¿Nodejs? ¿Apache? ¿Electron?

Debido a que debemos exponer nuestra aplicación a la interacción mediante una interfaz comprensible, es buena idea incluir los primeros 3 pasos (definición, prototipado e investigación) de la interfaz gráfica antes de comenzar el desarrollo.

3.4 Ejecución de reglas lógicas: desarrollo

¡Tiempo de programar!

Momento de configurar nuestro entorno de desarrollo para darle vida a nuestro software.

Es buena idea desarrollar el software incrementalmente, empezando por nuestro prototipo.

Por ejemplo, que la función “detectar_ciclista”, al principio, no detecte un ciclista, si no, que sólo devuelve una imagen de un ciclista que bajamos de internet.

Esto es para comprobar la interacción entre nuestras reglas a nivel de software.

Una vez finalizado el flujo completo de la aplicación (que la aplicación empiece y termine), podemos, incrementalmente, desarrollar las funcionalidades, ahora si, detectando al ciclista desde la cámara.

3. Exposición a una interfaz interactuable y comprensible

3.1. Definición
3.2. Prototipado
3.3. Investigación
3.4. Desarrollo

3. Exposición a una interfaz interactuable y comprensible

Los campos de estudio que se dedican a la usabilidad y el agrado de las aplicaciones son UI/UX.
UI (user interface, o interfaz del usuario) es referente a la composición de los elementos gráficos y la interacción de los mismos (colores, formas, animaciones, posicionamiento, tipografías). Mientras que UX (user experience, o experiencia del usuario) estudia la experiencia (valga la redundancia) del usuario con la aplicación, esto puede ser aplicado en qué palabras comunicar algo, qué días emitir cierta notificación, etc. Está más apuntado a la experiencia emocional del usuario en nuestra aplicación.

https://xd.adobe.com/ideas/process/ui-design/ui-vs-ux-design-understanding-similarities-and-differences/

Para que nuestra aplicación experimente el ser utilizada, la misma debe tener una interfaz comprensible. Esta interfaz puede tomar infinitas formas, ser linda o fea, cómoda o incómoda, pero principalmente debe ser comprensible.

Junto con la interfaz, podemos adjuntar un manual de su uso, lo que ayudaría, pero es esperado que la interfaz se explique a si misma, que “vaya llevando” al usuario.

3.1. Exposición a una interfaz interactuable : definición

¿Qué necesitamos del usuario? ¿Qué necesitamos mostrarle al usuario? Cuanto más reducidas sean estas respuestas, mejor.

No queremos que para el usuario sea tedioso usar nuestra aplicación.

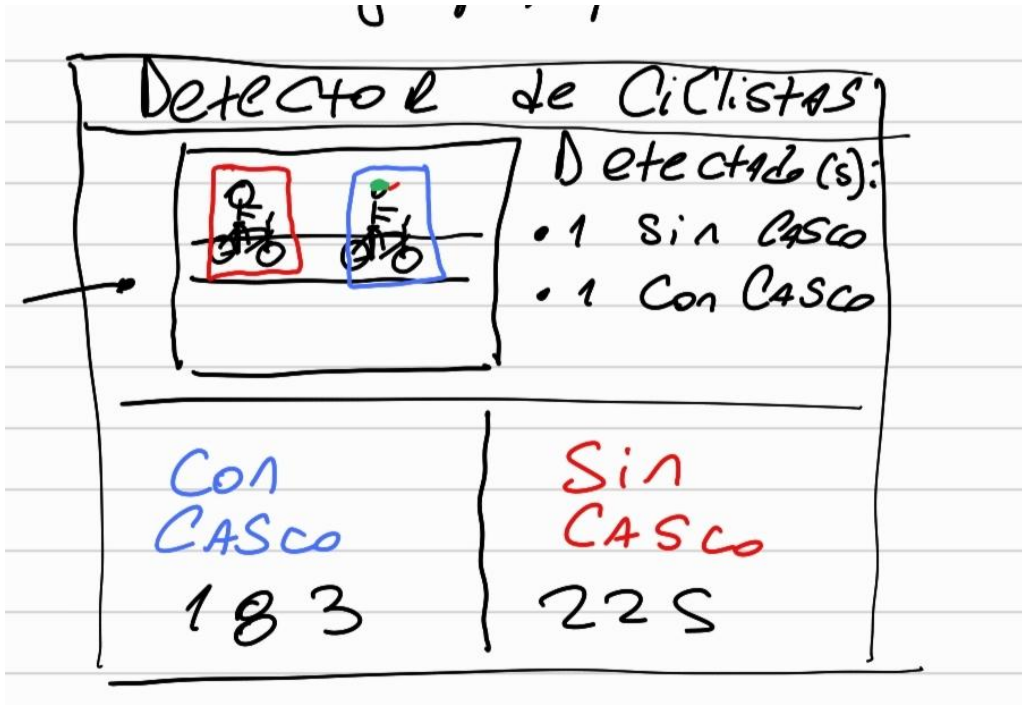
Por ejemplo, si tuviéramos un formulario donde llenar los datos personales, tener el campo “nombre” y “apellido”, puede ser más incómodo que tenerlos juntos.

Cuando mostramos un dato, tiene que ser conciso y fácil de digerir. Queremos reducir al máximo el trabajo del usuario, tanto para la entrada de información como para la salida.

3.2 Exposición a una interfaz interactuable: Prototipado

Tratándose de una interfaz gráfica, tiene sentido que el prototipo también lo sea.

Para nuestro ejemplo, podría ser:



3.3 Exposición a una interfaz interactuable: investigación

Podemos usar humanos reales para mostrarles nuestro prototipo y preguntarles si la interfaz es visualmente cómoda, agradable, entendible.

Es buena idea tener una opinión ajena cuando se trata de diseño.

Una alternativa es buscar diseños de aplicaciones ya existentes.

También, basándonos en UI/UX, podemos buscar en internet.

- Cómo mostrar números según UI/UX
- Colores recomendados según UI/UX
- Estructuras recomendadas para una aplicación según UI/UX
- Fuentes maś recomendadas (según UI/UX)
- etc..

3.4 Exposición a una interfaz interactuable: desarrollo

De manera muy similar al desarrollo de la aplicación de reglas lógicas, configuraremos nuestro ambiente de desarrollo y escribiremos código que, traducido al idioma que el ejecutor interpretará, haga que nuestra interfaz sea un producto utilizable.

Finalmente integraremos nuestra aplicación a la interfaz gráfica (por lo que es buena idea que estén bajo un mismo ambiente de desarrollo, que compartan el lenguaje de programación y formen parte de la estructura del proyecto)

???


Detector de ciclistas con casco
Category Bicycle
Con casco
0
Sin casco
1

Profit