GDPR Guida per sviluppatori

La CNIL pubblica una GDPR guida per sviluppatori

CNIL.
COMMISSION NATIONALE
INFORMATIQUE & LIBERTÉS

GDPR Guida per sviluppatori

Al fine di aiutare gli sviluppatori di applicativi e di siti web a rendere il loro lavoro rispondente ai requisiti del GDPR, la CNIL ha redatto una nuova guida alle buone pratiche sotto licenza aperta, in modo che possa essere arricchita da altri professionisti.

Questa versione italiana è stata gentilmente fornita da collaboratori esterni e rivista dal Garante per la Protezione dei Dati Personali. La *versione francese* è la versione autentica di questa guida.

La guida contiene indicazioni e buone pratiche, e pertanto fornisce a ogni stakeholder delle chiavi utili per comprendere il GDPR, quale che sia la dimensione della sua organizzazione. La guida può anche stimolare la discussione e lo sviluppo di pratiche all'interno delle organizzazioni e nelle relazioni con i clienti.

Che cosa contiene la guida?

Questa guida è divisa in **16 schede tematiche** che coprono la maggior parte delle necessità degli sviluppatori in ciascuno stadio di progetto, dalla preparazione dello sviluppo all'uso di analytics.

Il Regolamento Generale per la Protezione dei Dati Personali (GDPR) specifica che la protezione dei diritti e delle libertà delle persone fisiche richiede **"l'adozione di misure tecniche e organizzative adeguate per garantire il rispetto delle disposizioni del presente regolamento"** (Considerando 78).

La determinazione di tali misure è necessariamente **collegate al contesto delle operazioni di trattamento poste in essere**, e il Titolare del trattamento (l'entità pubblica o privata che tratta i dati personali) deve pertanto assicurare la protezione dei dati che è chiamato a trattare.

Le buone pratiche di questa guida, pertanto, **non intendono coprire tutte le richieste del Regolamento né intendono essere prescrittive**, ma forniscono un primo livello di misure per affrontare i problemi di sicurezza dei dati negli sviluppi IT che riguardano il trattamento di dati personali. A seconda della natura dei trattamenti, in alcuni casi potrà essere necessario implementare misure aggiuntive al fine di rispondere pienamente ai requisiti di legge.

Sommario

Come posso contribuire a questa guida?

Questa guida è disponibile in due versioni:

- Una *versione web sul sito di CNIL* e *nel tab "Releases"* di questo repository;
- Questa *versione GitHub*, che offre a chiunque la possibilità di contribuire.

Il contributo si articola in alcuni passi:

- Registrati su GitHub;
- Vai alla pagina di progetto
- Puoi:
  - usare il tab "Issue" per aprire commenti o partecipare alla discussione
  - Usare l'opzione "Fork" per apportare le tue modifiche e proporre la loro inclusione tramite il bottone "Pull Requests".

Le tue proposte di contributo verranno esaminate da CNIL prima della pubblicazione. La versione web della Guida al GDPR per sviluppatori sarà aggiornata regolarmente.

Uso

Per rilasciare tu stesso una copia di questo repository, puoi usare **Pandoc**. Questo strumento ti aiuta a convertire le schede in un unico documento docx, odt o HTML.
Puoi trovare le istruzioni su come installare Pandoc *qui*

- **Per generare un file .docx**:

```
pandoc -s --toc --toc-depth=1 -o Guide_GDPR_sviluppatori.docx [0-9][0-9]*.md
```

- **Per generare un file .odt**:

```
pandoc -s --toc --toc-depth=1 -o Guide_GDPR_sviluppatori.odt [0-9][0-9]*.md
```

- **To generare un file .html file**:

```
pandoc -s --template="templates/mytemplate.html" -H templates/pandoc.css -o index.html README.md [0-9][0-9]*.md
```

Scheda n°0: Sviluppa rispondendo al GDPR

Sia che tu lavori da solo, che tu sia parte di un gruppo di progetto, che tu gestisca un gruppo di sviluppatori, o che tu sia un fornitore di servizi che sviluppa per conto terzi, è essenziale che tu ti assicuri che i dati degli utenti e tutti i trattamenti di dati personali siano sufficientemente protetti durante tutto il ciclo di vita del progetto.

Questi passi ti aiuteranno a sviluppare applicativi e siti web che siano privacy-friendly:

1. **Sii consapevole dei principi cardine del GDPR**. Se lavori in un team, raccomandiamo che tu identifichi una persona responsabile di monitorare la compliance. Se la tua azienda ha un RPD/DPO (Responsabile della Protezione Dati/Data Protection Officer) allora quella persona è una risorsa fondamentale per *comprendere e rispondere agli obblighi del GDPR*. La nomina di un RPD in alcuni casi potrebbe essere obbligatoria, per esempio se i tuoi programmi o le tue app trattano dati cosiddetti "particolari" (vedi *esempi*) su larga scala o se effettuano monitoraggi regolari e sistematici su larga scala o se lavori per una pubblica amministrazione.
2. **Mappa e categorizza i dati e i trattamenti nel tuo sistema**. Una mappatura accurata dei trattamenti effettuati dal tuo programma o app ti aiuterà a garantire che rispondano ai requisiti di legge. Mantenere un *registro delle attività di trattamento* (un esempio lo puoi trovare sul *sito di CNIL*) ti permette di avere una visione d'insieme di questi dati, e di individuare e prioritizzare i rischi associati. Peraltro, dati personali potrebbero essere presenti in posti inattesi come server log, file di cache, file Excel, ecc., o potrebbero essere archiviati in una quantità di luoghi diversi. Nella maggior parte dei casi, la tenuta di un registro di questo tipo è obbligatoria.
3. **Prioritizza le azioni necessarie**. Sulla base del registro delle attività di trattamento, identifica prima dello sviluppo le azioni necessarie per rispondere agli obblighi del GPDR e prioritizza i punti di attenzione relativi ai rischi per gli interessati dal trattamento. Questi punti di attenzione riguardano in particolare *la necessità e i tipi di dati raccolti e trattati* dal tuo software, *le basi giuridiche* su cui si basano le tue operazioni di trattamento, *le informative* del tuo software o app, *le clausole contrattuali* che ti legano ai tuoi fornitori, i termini e le condizioni per *esercitare i diritti*, e le misure implementate per *la sicurezza dei trattamenti*.
4. **Gestisci i rischi**. Quando determini che un trattamento di dati personali può creare rischi elevati per gli interessati, assicurati di gestire i rischi in modo appropriato al contesto. Un *Privacy Impact Assessment (PIA)* può aiutarti a gestirli. La CNIL ha sviluppato un *metodo*, dei *modelli di documento* e un *tool* che ti aiuteranno a individuare i rischi nonché un *catalogo di buone pratiche* che ti assisterà nella implementazione delle misure per rispondere ai rischi che avrai identificato. Inoltre, un Privacy Impact Assessment è obbligatorio per tutti quei trattamenti che possono creare rischi gravi peri diritti e le libertà degli interessati. Lo CNIL propone, sul suo *sito*, un elenco dei tipi di trattamento per i quali un Privacy Impact Assessment è obbligatorio. Un analogo elenco è disponibile sul sito del Garante.
5. **Attiva dei processi interni** per garantire la compliance durante tutte le fasi dello sviluppo, assicura che ci siano procedure interne per garantire che la protezione dei dati sia tenuta in conto in tutti gli aspetti del progetto e a fronte di qualsiasi evento possa verificarsi (ad es. falla di sicurezza, risposta a richieste di rettifica o di accesso, modifica dei dati raccolti, cambio di fornitore, trafugamento di dati, ecc.). I requisiti alla base dell'*etichetta di governance* (anche se non viene più rilasciata da CNIL dopo l'entrata in vigore del GDPR) possono costituire una utile base per aiutarti a definire le necessarie misure organizzative.
6. **Documenta la compliance dello sviluppo** per dimostrare in ogni momento la tua compliance al GDPR: ci deve essere piena consapevolezza delle azioni compiute e dei documenti prodotti a ciascuno stadio dello sviluppo. Questo implica in particolare la periodica revisione e

l’aggiornamento della documentazione così da renderla sempre consistente con le caratteristiche del tuo programma.

Il sito di CNIL fornisce numerosi esempi pratici che ti assisteranno, a seconda del tuo settore di attività, nella definizione di trattamenti dati rispondenti alla legge.

Scheda n°1: Individua i dati personali

Comprendere i concetti di "dato personali", "scopo" e "trattamento" è essenziale per sviluppare trattamenti rispondenti ai requisiti di legge. In particolare, stai attento a non confondere "anonimizzazione" e "pseudonimizzazione", che sono definiti con molta precisione nel GDPR.

Definizione

- Il concetto di **dato personale** è definito nel *Regolamento Generale per la Protezione dei Dati Personali* come *"qualsiasi informazione riguardante una persona fisica identificata o identificabile («interessato»)"*. La definizione copre un ambito vasto che include tanto dati che identificano direttamente (ad es. nome e cognome) quanto dati che identificano indirettamente (ad es, numero di telefono, targa dell'auto, identificativo del terminale, ecc.).
- Qualsiasi operazione su dati di questo tipo (raccolta, registrazione, trasmissione, modifica, disseminazione, ecc.) costituisce **trattamento ai sensi dei GDPR** e deve quindi rispondere ai requisiti definiti dalla legge. Queste operazioni di trattamento devono essere lecite e avere una finalità specificata. I dati personali raccolti e trattati devono essere rilevanti e limitati a quelli strettamente necessari per raggiungere la finalità prefissata.

Esempi di dati personali

- Quando si riferiscono a persone fisiche, **i seguenti dati sono dati personali**:
  - Cognome, nome, pseudonimo, data di nascita
  - foto e registrazioni della voce
  - numero di telefono fisso o mobile, indirizzo postale, indirizzo email
  - indirizzo IP, identificatore di connessione telematica o cookie identificativo
  - Impronta digitale, impronta del palmo o reticolo venoso della mano, impronta retinica
  - Targa automobilistica, codice fiscale, numero di matricola
  - Dati di utilizzo dell'applicazione, commenti, ecc.
- **L'identificazione di persone fisiche si può ottenere**:
  - da un singolo dato (esempi: cognome e nome)
  - dall'incrocio di insiemi di dati (esempio: una donna che vive a un dato indirizzo, nata in una certa data e membro di una data associazione).

- Alcuni dati sono considerati **particolarmente sensibili**. Il GDPR proibisce la raccolta o l'uso di questi dati a meno che, in particolare, l'interessato abbia dato il proprio consenso (consenso attivo, esplicito, libero, specifico e informato).

- I requisiti di cui al punto precedente riguardano i seguenti tipi di dati:

  - dati relativi alla **salute degli individui**
  - dati relativi alla **vita sessuale** o **all'orientamento sessuale**
  - dati che possono rivelare una (anche presunta) origine **razziale** o **etnica**
  - opinioni politiche, credo religiosi, convinzioni filosofiche o appartenenza a sindacati
  - dati **genetici** e **biometrici usati allo scopo di identificare in modo univoco un individuo**.

Anonimizzazione di dati personali

- Un **processo di anonimizzazione di dati personali** mira a rendere impossibile, a meno di uno sforzo ritenuto irragionevole, l'identificazione degli individui appartenenti al dataset. Quando questa

anonimizzazione riesce, i dati non sono più considerati dati personali e le prescrizioni del GDPR non si applicano più.

- Di default, ti raccomandiamo di **non considerare mai anonimi dei dataset sui quali non è stata applicata alcuna tecnica di anonimizzazione.** L'anonimizzazione è il risultato del trattamento di dati personali al fine di prevenire l'identificazione. Una tecnica di anonimizzazione deve evitare che i seguenti eventi possano verificarsi :
- *individuazione*: non è possibile isolare uno o tutti i record del dataset che identificano un individuo
  - *collegabilità*: il dataset non consente di collegare due o più record che si riferiscono a uno stesso interessato o a uno stesso gruppo di interessati
  - *inferenza*: non è possibile dedurre, con probabilità significativa, il valore di un attributo a partire dai valori di un insieme di altri attributi.

- queste operazioni di trattamento implicano nella maggior parte dei casi una **perdita di qualità del dataset risultante**. L'*opinione sulle tecniche di anonimizzazione* dell Gruppo di Lavoro Articolo 29 (Art. 29 WP) descrive le principali tecniche di anonimizzazione attualmente in uso, nonché esempi di dataset erroneamente considerati anonimi. È importante notare che le tecniche di anonimizzazione hanno delle controindicazioni. La scelta se anonimizzare o meno i dati, nonché la scelta della particolare tecnica di anonimizzazione deve essere fatta caso per caso, sulla base del particolare contesto, dell'uso e delle necessità (natura dei dati, utilità dei dati, rischi per le persone, ecc.) .

Pseudonimizzazione di dati personali

- **La pseudonimizzazione è un compromesso fra mantenere i dati nella loro forma originale e produrre dataset anonimizzati**.

- Si riferisce al trattamento di dati personali in modo che **dati che si riferiscono a una persona fisica non possano più essere attribuiti ad essa senza l'uso di informazioni aggiuntive**. Il GDPR rimarca che tali informazioni aggiuntive debbano essere conservate separatamente e soggette a misure tecniche e organizzative volte a evitare la re-identificazione degli interessati. La pseudonimizzazione può essere un processo reversibile.

- In pratica, un processo di pseudonimizzazione consiste nel **sostituire in un dataset i dati immediatamente identificanti (cognome, nome, ecc.) con dati indirettamente identificanti** (pseudonimo, numero di pratica, ecc.) allo scopo di ridurre il loro potere identificativo. Questi dati indiretttamente identificativi potrebbero essere il risultato di un hash crittografico di dati degli personali, come l'indirizzo IP, la user ID, l'indirizzo e-mail.

- I dati risultanti dalla pseudonimizzazione sono considerati come **dati personali e pertanto soggetti alle prescrizioni del GDPR**. Ad ogni modo, il Regolamento Europeo incoraggia l'uso della pseudonimizzazione nel trattamento di dati personali. Inoltre il GDPR ritiene che la pseudonimizzazione renda possibile ridure il rischio per gli interessati e contribuisca alla compliance con il Regolamento.

Scheda n°2: Prepara lo sviluppo

I principi della protezione dei dati personali devono essere integrati negli sviluppi IT dalla fase di design in poi, al fine di proteggere la privacy delle persone i cui dati verranno trattati, per dare loro un maggior controllo sui loro dati e per limitare errori, perdite, modifiche non autorizzate o abusi dei loro dati nelle applicazioni.

Scelte metodologiche

- **Metti la protezione della privacy al centro dei tuoi sviluppi** adottando una metodologia di *Privacy By Design*.
- Considera di **integrare la sicurezza al centro dei processi**. ANSSI rende disponibile una guida *"digital security & agility"* (solo in Francese) che indica come portare avanti gli sviluppi nell'ambito di un framework agile tenendo in conto gli aspetti di sicurezza. Prendila come fonte di ispirazione.
- Per ogni sviluppo rivolto al grande pubblico, **considera i setting di sicurezza** e in particolare i valori di default come, per esempio, le caratteristiche e i contenuti utente visibili per default.
- **Conduci un *Privacy Impact Assessment (PIA)***. Per *alcune operazioni di trattamento* è obbligatorio. In altri casi costituisce una buona pratica che ti permetterà di individuare e affrontare i rischi a valle del tuo sviluppo. La CNIL ha una sezione speciale del proprio sito e fornisce un *software libero* dedicato a questo tipo di analisi.

Scelte tecnologiche

Architettura e caratteristiche

- **Includi la protezione della privacy, inclusi i requisiti per la sicurezza dei dati, nella fase di design dell'applicazione o del servizio.** Questi requisiti dovrebbero influenzare le *scelte architetturali* (ad es. decentralizzato vs. centralizzato) o le funzionalità (ad es. anonimizzazione, minimizzazione dei dati). Le regolazioni di default dell'applicazione devono almeno rispondere ai requisiti minimi di sicurezza e rispondere ai requisiti di legge. Per esempio, la complessità di default delle password deve rispondere come minimo alle *raccomandazioni CNIL sulle password*
- **Mantieni il controllo del tuo sistema**. È importante che tu mantenga il controllo del tuo sistema, sia per assicurarne il corretto funzionamento, sia per garantire un alto livello di sicurezza. Mantenere il tuo sistema semplice ti permette di comprendere con precisione come funziona e di individuarne i punti deboli. Se una certa complessità è necessaria, è consigliabile partire da un sistema semplice, sicuro e progettato correttamente. Su questa base è possibile aumentare la complessità passo a passo, mettendo via via in sicurezza le nuove funzionalità che vengono aggiunte.
- **Non contare su una sola linea di difesa**. Nonostante tutte le misure prese per progettare un sistema sicuro, può sempre accadere che alcune componenti aggiunte in un secondo tempo non siano sufficientemente sicure. Per minimizzare i rischi per gli utenti finali, è suggeribile che il sistema adotti una difesa in profondità. Per esempio, controllare i valori immessi in un modulo online è parte delle difese perimetrali. Se questa difesa viene scavalcata, la protezione delle query al database può risultare compromessa.

Strumenti e pratiche

- **Usa standard di programmazione che tengono in conto la sicurezza**. Spesso sono già disponibili liste di standard, buone pratiche o guide di programmazione per migliorare la sicurezza dei tuoi sviluppi. Puoi anche integrare tool aggiuntivi nel tuo ambiente integrato di sviluppo ("**IDE**") in modo da controllare in automatico che il tuo codice corrisponda alle regole

dettate dagli standard e dalle buone pratiche. Su Internet puoi facilmente reperire elenchi di buone pratiche per i tuoi linguaggi di programmazione preferiti. Per esempio *qui* per C, C++ o Java. Esistono anche buone pratiche specifiche per lo sviluppo di applicazioni Web, come quelle pubblicate da *OWASP*.

- **Le scelte tecnologiche sono critiche**. Alcuni aspetti che devi tenere in conto:
  - A seconda del campo di applicazione o della funzionalità sviluppata, uno specifico linguaggio o tecnologia potrebbe essere più appropriato di un altro.
  - I linguaggi e le tecnologie più maturi sono più sicuri. In generale, sono stati soggetti a audit per correggere le vulerabilità più note. Ad ogni modo, devi fare attenzione a usare le ultime versioni di ciascun componente tecnologico che userai.
  - Devi evitare di scrivere la tua soluzione in un linguaggio che hai appena imparato e non domini ancora completamente. La tua mancanza di esperienza potrebbe esporti a maggiori rischi.
- **Appronta un ambiente di sviluppo sicuro che ti consenta la gestione delle versioni del codice** seguendo la *scheda dedicata* di questa guida.

Scheda n°3: Sviluppa in un ambiente sicuro

La sicurezza dei server di produzione, sviluppo e continuous integration, non ché quella delle macchine degli sviluppatori deve essere una priorità, perché centralizzano l'accesso a vaste moli di dati.

Valuta i tuoi rischi e adotta misure di sicurezza appropriate

- **Valuta i rischi** degli strumenti e dei processi che usi per lo sviluppo. Fai un inventario delle misure di sicurezza esistenti e definisci un piano d'azione per migliorare la tua copertura dai rischi. Incarica una persona come responsabile della sua implementazione.
- Considera i rischi per tutti gli strumenti che usi, inclusi gli strumenti SaaS (Software as a Service) e gli strumenti collaborativi in cloud (come *Slack*, *Trello*, *GitHub*, ecc.).

Metti in sicurezza server e workstation in modo omogeneo e riproducibile

- Una lista di raccomandazioni riguardo alla sicurezza di server, workstation e reti interne sono disponibili nelle *schede dal n° 5 all'8* della Guida alla Sicurezza dei dati personali della CNIL.
- **Redigi un documento che elenchi le misure di sicurezza e ne spieghi la configurazione**, per assicurare che le misure di sicurezza siano implementate uniformemente sia sui server che sulle workstation. Per ridurre il carico di lavoro, puoi usare **strumenti di configuration management** come *Ansible*, *Puppet* o*Chef*.
- Aggiorna sempre server e workstation, se possibile in modo automatico. Puoi definire una watchlist delle vulnerabilità più importanti, per esempio gli *NVD Data Feed* del NIST.

Poni particolare attenzione alla gestione degli accessi e alla tracciabilità delle operazioni

- Ricordati di documentare la gestione delle tue **chiavi SSH** (uso di algoritmi di crittografia e lunghezze delle chiavi allo stato dell'arte, protezioni delle frasi con una password, rotazione delle chiavi). Per esempi di buone pratiche, vedi il *documento sull'uso siduro di (open)SSH*.
- Incoraggia l'autenticazione forte nei servizi usati dal team di sviluppo.
- **Traccia** gli accessi alle macchine e, se possibile, implementa una **analisi automatica dei log**. Per poter raccogliere tracce affidabili, occorre evitare l'uso di account generici.

Scheda n°4: Gestisci il codice sorgente

Qualunque sia la dimensione del tuo progetto, ti raccomandiamo fortemente l'uso di uno strumento di gestione del codice sorgente, come ad esempio un *version control system*, per tenere traccia delle differenti versioni nel corso del tempo.

Definisci un sistema efficiente di controllo delle versioni efficiente, pensando alla sua sicurezza.

- Un sistema di controllo delle versioni è un programma software che ti permette di archiviare **tutto il tuo codice sorgente e i file associati** mantenendo la **cronologia di tutte le modifiche** che sono state fatte. Un semplice server FTP non è un sistema di controllo delle versioni.
- Imposta il tuo ambiente correttamente usando le funzionalità offerte dal tuo sistema di controllo delle versioni. Ti raccomandiamo di utilizzare una **autenticazione forte** o un'**autenticazione con chiavi SSH** sin dall'inizio del tuo progetto.
- Oltre a questo, assegna dei *livelli di accesso* al progetto per i diversi utenti del tuo sistema di controllo delle versioni e per ciascun livello definisci i corrispondenti **permessi** (per esempio, un livello "guest" con diritti di limitati di lettura, un livello "developer" con diritti di scrittura, ecc.)
- Fai **backup** regolari del tuo sistema di gestione del codice sorgente: In particolare, ricordati di fare il backup del server principale dove vengono registrate tutte le modifiche.
- Definisci procedure di sviluppo per lavorare in efficienza anche se **più persone sviluppano in contrmporanea**. Per esempio, puoi decidere **che non tutti lavorino sullo stesso ramo (*master branch*) ma che ciascuno lavori su un ramo distinto, che verrà poi inglobato nel ramo principale mano a mano che lo sviluppo procede. Questo tipo di strategie sono già ben documentate, per esempio in *Git Flow*.** In aggiunta, alcuni sistemi di controllo delle versioni ti permettono di definire **branch protetti** che impediscono cambiamenti non autorizzati ai file di quei branch.

Sii consapevole del contenuto del tuo codice sorgente.

- Adotta **strumenti di valutazione della qualità del codice** (quality metrics) che analizzano il codice al momento del *commit* per verificarne la buona qualità. Puoi anche aggiungere script per il controllo di queste metriche alla *configurazione del sistema di controllo delle versioni*, in modo che il *commit* sia annullato se il sorgente non è di una qualità sufficiente.
- Mantieni i file segreti e le password di fuori del repository del codice sorgente:
  - in **file separati, non soggetti a *commit***. Ricorda di usare i file speciali del tuo sistema di controllo delle versioni (come ad esempio *.gitignore* PER *Git*) in modo che non ti succeda per errore di fare *commit* di file sensibili
  - riguardo alle **variabili d'ambiente** assicurati di controllare che non vengano accidentalmente registrate nei *log* o visualizzate quando un applicativo va in errore
  - usa ***specifici software di configuration management o di secret management***.
- Infine, se devi includere dati di questo genere nel tuo repository, prendi in considerazione la possibilità di **cifrare/decifrare automaticamente** questi file tramite un *plugin* del sistema di controllo di versione (ad es. git-crypt).
- Dopo un *commit* che contiene dati personali o altri dati critici, non dimenticare di fare un purge *completo* del repository del codice sorgente: anche dopo ulteriori modifiche, quei dati potrebbero essere ancora disponibili nella *history* del tuo repository.

- Fa' attenzione prima di **pubblicare online il tuo codice sorgente**. Passa in rassegna **tutti i contenuti**, inclusa tutta la *history* delle modifiche.

Esempi di strumenti

- A differenza di strumenti come *Subversion*, che richiedono un server centrale per funzionare, i principali sistemi di controllo delle versioni (come ad esempio*Git* o *Mercurial* sono **decentralizzati**.
- La maggior parte di questi sistemi, inclusi gli strumenti relativi (bug management, wiki per la documentazione, ecc.), sono anche disponibili attraverso **un'interfaccia web** . Queste soluzioni possono essere accessbili via Internet (*GitHub*, *Bitbucket*, ecc.), o possono essere integrate nei tuoi server.

Scheda n°5: Fai scelte architetturali informate

Quando progetti l'architettura della tua applicazione, devi identificare i dati personali che verranno raccolti, e definire un percorso e un ciclo di vita per ciascuno di essi. La scelta degli asset di supporto (storage locale, server servizi in cloud) è un momento cruciale, che deve essere adeguato ai tuoi bisogni ma anche alla tua conoscenza tecnica. Il registro dei trattamenti e la conduzione di un Privacy Impact Assessment ti possono assistere in questa scelta.

Esaminare il ciclo di vita di dati e processi, dalla raccolta alla cancellazione.

- Descrivi e rappresenta il modo in cui funziona il tuo prodotto prima di avviare il progetto, usando diagrammi di tipo *data flow* e fornendo una descrizione dettagliata dei processi.
- Quando i dati sono **registrati solo sul terminale dell'utente** (storage locale) o **rimangono confinati a reti sotto il controllo dell'utente** (ad es. Wi-Fi o altra rete locale), la sicurezza dei dati è il principale punto di attenzione. Il periodo di conservazione dei dati e la loro cancellazione dovrebbero essere determinati dai singoli utenti.
- **quando i dati viaggiano attraverso servizi online**, devi scegliere se a avvalerti di un servizio di hosting o usare un fornitore di servizi sulla base delle tue competenze di sicurezza e della qualità che ci si aspetta dal servizio. I fornitori cloud più noti possono offrire livelli di sicurezza più elevati, ma generano nuovi rischi che devono essere tenuti sotto controllo. Queste *Raccomandazioni per le aziende che desiderano avvalersi di servizi di cloud computing* possono guidarti durante questo stadio di scelta.

Nel caso si usi un hosting esterno

- **Scegli un fornitore di servizi che ti assicuri misure adeguate di sicurezza e confidenzialità e sia sufficientemente trasparente**.
- **Assicurati di conoscere la collocazione geografica dei server che accoglieranno i tuoi dati**. Ti potrebbe venire chiesto di trasferire i dati al di fuori dell'Unione Europea (UE) e della European Economic Area (EEA). Mentre i dati possono muoversi liberamente all'interno della UE/EEA, i trasferimenti al di fuori idi UE/EEA sono possibili solo se vengono assicurati livelli sufficienti e appropriati di protezione dei dati personali. Sul sito di CNIL puoi trovare una mappa che mostra *i diversi livelli di protezione dei dati nei diversi paesi del mondo*.
- **Se devi immagazzinare dati relativi alla salute**, assicurati che il provider di cui ti servi sia *certificato* o *approvato* per questo genere di attività.
- Altri punti da tenere presente:
  - l'esistenza di una policy di sicurezza accessibile;
  - misure fisiche di protezione e sicurezza presso il sito di hosting;
  - cifratura dei dati e altri processi per assicurare che il provider non abbia accesso ai dati che gli vengono affidati;
  - gestione degli aggiornamenti, gestione delle autorizzazioni, autenticazione del personale e sicurezza degli sviluppi applicativi;
  - facilità della reversibilità/portabilità dei dati in un formato strutturato di uso comune, in qualsiasi momento e su richiesta.

Scheda n°6: Metti in sicurezza siti, applicazioni e server

Ogni sito, applicazione e server deve incorporare regole di base di sicurezza allo stato dell'arte, non solo per le comunicazioni di rete, ma anche per l'autenticazione e la gestione dell'infrastruttura.

Mettere in sicurezza le reti di comunicazione

- **Implementare TLS versione 1.2 or 1.3** (al posto di SSL) su tutti i siti web e per le trasmissioni dati elle tue applicazioni mobili, per esempio con *LetsEncrypt*, usando solo le versioni più recenti e controllando la correttezza dell'implementazione.
- **Rendi obbligatorio l'uso di TLS** per tutte le pagine del tuo sito e per tutte le applicazioni mobili.
- **Riduci le porte i comunicazione aperte** a quelle strettamente necessarie per il corretto funzionamento delle applicazioni installate. Se l'accesso a un server web è possibile solo attraverso il protocollo HTTPS, allora solo le port 80 e 443 del server devono essere accessibili, e tutte le altre porte possono essere bloccate dal firewall.
- **OWASP ha pubblicato delle schede-guida**, ad esempio per *implementare correttamente TLS* o per *mettere in sicurezza un webservice*.

Mettere in sicurezza le autenticazioni

- **Segui le *raccomandazioni CNIL per le password***. In particolare, ricordati di mettere un limite al numero di tentativi di accesso.
- **Non archiviare mai le passowrd in chiaro**. Memorizza il loro hash usando una libreria consolidata, come *bcrypt*.
- **Se usi cookie per l'autenticazione**, ti raccomandiamo:
- di forzare l'uso di HTTPS tramite *HSTS*;
- di usare il flag `secure`;
- usa il flag `HttpOnly`.
- **Testa le librerire di crittografia installate sui tuoi sistemi** e disabilita quelle obsolete (RC4, MD4, MD5 etc.). Incoraggia l'utilizzo di AES256. *Leggi le note di OWASP al riguardo*.
- **Adotta una politica specifica per le password degli amministratori**. Come minimo, cambia le loro password ogni volta che un amministratore lascia il lavoro, e comunque sempre in caso di sospetta violazione della sicurezza.
- **Limita l'accesso ai tool e alle interfacce di amministrazione al solo staff qualificato per il loro uso.** Per le operazioni quotidiane, incoraggia l'utilizzo di account a privilegi ridotti.
- **L'accesso remoto alle interfacce di amministrazione deve essere soggetto a misure di sicurezza rinforzate.** Per esempio, per i server interni, può essere una buona soluzione dotarsi di una VPN con autenticazione forte dell'utente e della macchina che usa per connettersi.

Mettere in sicurezza le infrastrutture

- **Fai backup regolari, cifrati e controllati regolarmente**. Questo è particolarmente utile nel caso di attacchi di tipo *ransomware* perché in quel caso la disponibilità di un backup di tutti i sistemi è la tua sola possibilità di ripristinare i sistemi.
- **Limita la dimensione dello stack software che impieghi** e per ciascun elemento dello stack:

- **Installa gli update critici** senza ritardo, programmando un controllo automatico settimanale;
  - **Automatizza il controllo delle vulnerabilità** abbonandoti per esempio ai *Data Feed di NVD*.
- **Adotta strumenti di scoperta delle vulnerabilità** per i processi più critici, in modo da poter scoprire possibili violazioni di sicurezza. Puoi usare sistemi per la scoperta e la prevenzione di attacchi anche sui sistemi e sui server critici. Questi test devono essere condotti regolarmente e comunque prima della messa in produzione di ogni nuovo software.
- **Limita o proibisci l'accesso sia fisico che via software alle porte di diagnostica e di configurazione remota**. Per esempio, puoi avere l'elenco di tutte le porte aperte con lo strumento `netstat`.
- **Proteggi i database che esponi su Internet**, come minimo limitando il più possibile l'accesso e cambiando la password di default per l'account dell'amministratore.
- Per quanto riguarda la gestione di database, le buone pratiche includono:
  - per l'accesso al database **usare account nominativi** e creare account specifici per ciascuna applicazione;
  - **revoca i privilegi di amministratore** degli account (utente o applicativo) per evitare modifiche alla struttura del database (tabelle, viste, processi, ecc.);
  - assicurati di proteggerti contro attacchi di tipo SQL injection o *script injection;
  - incoraggia la cifratura a riposo di dischi e database.

Scheda n°7: Minimizza la raccolta di dati

Devi raccogliere solo i dati personali rilevanti, adeguati e necessari per la finalità per la quale li raccogli, come definita al momento della raccolta.

Prima della raccolta, pensa ai diversi tipi di dati che devi raccogliere e limita la raccolta a ciò che è strettamente necessario.

- Pensa ai diversi **tipi di dati** che dovrai raccogliere prima di implementare la tua applicazione e **documenta** questi ragionamenti.
- Se **per una certa categoria di persone non servono** alcuni dati, non raccoglierli.
- Tratta e archivia i dati in modo da generalizzarli (analogamente a quanto avviene per la pseudonimizzazione). Per esempio, se l'applicazione necessita solo dell'anno di nascita, archivia soltanto l'anno di nascita invece dell'intera data di nascita.
- Se raccogli dati particolarmente sensibili, come dati riguardanti la salute o informazioni giudiziarie, assicurati di raccogliere solo il **minimo necessario**. A causa delle restrizioni legali, la soluzione più semplice è ancora una volta **non raccoglierli** se puoi farne a meno.
- Minimizza anche i dati raccolti nei **log di sistema** e non fargli registrare dati sensibili o critici (password, dati sanitari, ecc.)
- Alcune funzionalità potrebbero migliorare l'esperienza utente ma **non sono strettamente necessarie per il corretto funzionamento dell'applicazione** (ad es. la geolocalizzazione per migliorare una ricerca geografica). In questo caso, l'utente finale deve poter **scegliere se usare o meno queste funzionalità**. Se decide di usarle, i dati ulteriori che raccoglierai devono essere conservati solo per il tempo strettamente necessario e non devono mai essere usati per altri scopi.
- Ricordati di associare un **periodo di conservazione** a ciascuna categoria di dati, sulla base dello scopo del trattamento e degli obblighi di legge relativi alla loro conservazione. Anche i log di sistema devono avere un periodo di conservazione predefinito. Documenta i periodi di conservazione che hai definito, perché dovrai poterli motivare.

Una volta che i dati sono stati raccolti, predisponi meccanismi di cancellazione automatica.

- Implementa un sistema automatico di **eliminazione** al termine della conservazione. Puoi anche predisporre revisioni manuali periodiche dei dati conservati.
- Per assicurare una cancellazione completa, cancella **fisicamente** tutti i dati non più necessari tramite strumenti specializzati, o distruggendo i supporti fisici.
- Se i dati sono ancora utili, puoi ridurne il potere identificativo usando metodi di **pseduonimizzazione** o **anonimizzazione**. Nel caso della pseudonimizzazione, i dati rimangono soggetti alle norme sulla protezione dei dati personali (vedi *Scheda 1*).
- Tieni un log delle **procedure di cancellazione automatica**. Questi log possono essere usati come **prova di cancellazione** di un certo dato.

Scheda n°8: Gestisci i profili utente

La gestione dei profili dello staff e degli utenti deve essere valutata a monte dello sviluppo. Si tratta di definire profili di accesso e autorizzazione differenti in modo che ciascun utente possa accedere solo ai dati di cui ha effettiva necessità.

Buone pratiche per la gestione degli utenti

- Il punto di partenza è l'**uso di identificativi unici e individuali**, tanto per gli utenti dell'applicazione che per il team di sviluppo.
- Assicurati di richiedere l'autenticazione prima di qualunque accesso a dati personali, in linea con le *raccomandazioni di CNIL*.
- Per assicurarti che ogni persona (utente o staff) possa accedere solo **ai dati di cui ha effettiva necessità**, il tuo sistema deve fornire **politiche differenziare di gestione degli accessi ai dati** (lettura, scrittura, cancellazione, ecc.) a seconda delle persone e delle loro necessità. Un meccanismo generale di gestione dei profili utente ti permetterà di raggruppare diritti diversi a seconda del ruolo svolto da un gruppo di utenti all'interno dell'applicazione.
- La gestione dei profili utente si può accompagnare a **sistemi di log che consentano di tracciare le attività e individuare anomalie o eventi legati alla sicurezza**, come accessi fraudolenti o uso improprio di dati personali. Questi sistemi non devono essere usati per altro scopo che non sia garantire un utilizzo corretto dei sistemi informatici. Inoltre, i log non possono essere conservati più di quanto sia necessario. In generale, si considera adeguata una conservazione per sei mesi.
- Considera anche di pianificare audit del codice o *penetration testing* per il tuo ambiente di sviluppo per **garantire la robustezza del tuo sistema di gestione dei profili utente**.

Semplifica la gestione dei profili di sicurezza

- Pianifica di **documentare o automatizzare gli spostamenti dello staff**. Per esempio, dovresti definire procedure che prescrivono cosa fare nel momento in cui una persona non è più autorizzata ad accedere a un locale o a una risorsa IT, o al termine del contratto.
- Gestire staff e utenti implica **una revisione periodica dei permessi** sulla base dell'evoluzione delle necessità e della mobilità organizzativa all'interno del progetto. L'uso di servizi di directory come ad esempio *Lightweight Directory Access Protocol*, ti aiuterà a tenere sotto controllo questi cambiamenti e ti permetterà di affinare le tue strategie di accesso, per esempio definendo ruoli sulla base dei profili di utilizzo delle risorse. Questo ti permetterà di rispettare al meglio il principio del minimo privilegio.
- L'uso di account "superutente" (come *root*, *Administrator*, ecc.) deve essere evitato per le operazioni quotidiane e di routine, perché costituiscono le fondamenta del tuo sistema e allo stesso tempo un bersaglio di elezione per un possibile attaccante esterno. In particolare per questi account ti raccomandiamo di adottare una politica di password forti (da 10 a 20 caratteri, oppure multi-fattore) e di limitare al massimo il numero di persone che le conoscono.
- **Favorisci l'utilizzo di un password manager all'interno del progetto** e, ogni volta che sia possibile, la transizione all'autenticazione forte. Evita account generici condivisi fra più persone.

Scheda n°09: Controlla librerie e SDK

Usi librerie, SDK o altri componenti software scritti da terze parti? Ecco alcuni suggerimenti su come integrarli mantenendo comunque il controllo del tuo sviluppo.

Fai una scelta informata

- **Valuta il valore di ciascuna dipendenza che aggiungi.** Alcuni componenti software di uso comune sono lunghi solo poche righe. Però, ogni elemento aggiunto significa un aumento della superficie di attacco del tuo sistema. Se una singola libreria offre più funzionalità, integra solo le funzionalità di cui hai reale esigenza. Attivando il numero minimo di funzionalità riduci il numero di potenziali *bug* a cui sarai esposto.

- **Scegli software, librerie e SDK che vengono sottoposti a revisioni:**

- Se vuoi usare software libero o open source, scegli progetti o soluzioni con una comunità attiva di utenti, aggiornamenti regolari e buona documentazione.

- Se usi altri tipi di soluzioni con un supporto commerciale, assicurati contrattualmente che il codice sia mantenuto e aggiornato per la durata di vita del tuo progetto.

- **Tieni in considerazione la privacy.** Alcune librerie e alcuni SDK si ripagano usando i dati personali raccolti dalle applicazioni e dai siti in cui sono integrati. Assicurati che queste terze parti rispondano alle leggi applicabili sui dati personali, inclusa la corretta raccolta del consenso.

- **Se usi meccanismi crittografici, ti sconsigliamo fortemente di implementarti da solo gli algoritmi di cifratura**, piuttosto scegli librerie crittografiche che sono riconosciute, mantenute e semplici da usare.

Valuta gli elementi individuati

- **Leggi la documentazione e cambia le configurazioni di default**. È importante sapere come funzionano le dipendenze del tuo codice. Le librerie di terze parti e gli SDK di solito hanno dei file di configurazione di default che spesso, per mancanza di tempo, non vengono nemmeno adattati, aprendo la via a molte falle di sicurezza.
- **Fai un audit delle tue librerie e degli SDK.** Sai davvero tutto quello che fanno le librerie e gli SDK che integri nel tuo sviluppo? Quali dati vengono trasmessi attraverso queste dipendenze, e a chi? Un audit ti permetterà di determinare le prescrizioni di legge sulla protezione dei dati a cui devi attenerti e di stabilire le responsabilità di tutti gli attori coinvolti. **Fai una mappa delle dipendenze.** Le librerie di terze parti possono anche integrare altre componenti: fare un audit del loro codice ti permetterà di mappare meglio le tue dipendenze e di agire al meglio se una di loro mostra dei problemi. Per le componenti di terze parti, raccomandiamo anche che tu faccia degli audit di sicurezza e che li tenga poi monitorati.
- **Fai attenzione al *typosquatting* e altre tecniche malevole.** Controlla i nomi delle dipendenze, e delle loro dipendenze, per evitare attacchi. Non fare copia-e-incolla di linee di comando da siti che non conosci.

Mantieni aggiornate librerie e SDK

- **Usa sistemi di gestione delle dipendenze** (come yum, apt, maven, pip, ecc.) per mantenere una lista aggiornata delle tue dipendenze.
- **Gestisci gli aggiornamenti delle tue dipendenze**, specialmente nel caso di aggiornamenti di sicurezza che risolvono delle vulnerabilità. Devi predisporre una procedura documentata per gestirle e metterle in produzione nel minor tempo possibile.

- **Fai attenzione a versioni di librerie e SDK a fine vita o a fine contratto** che non saranno più supportati: cerca di trovare un'altra soluzione (scegli una nuova libreria, rinnova il supporto commerciale).
- **Controlla lo status dei progetti open-source**, in particolare il cambio di dominio o di proprietà del pacchetto, alcuni attacchi usano aggiornamenti fasulli di dipendenze ampiamente in uso.

Scheda n°10: Garantisci la qualità del codice e della documentazione

È essenziale adottare il prima possibile buone tecniche di scrittura del codice. La leggibilità del codice riduce nel tempo gli sforzi di mantenimento e di correzione dei bug per te e per i tuoi (magari futuri) collaboratori.

Documenta il codice e l'architettura

- A volte durante lo sviluppo la documentazione viene trascurata, per mancanza di tempo o di visibilità del progetto. Però, è **cruciale per la manutenibilità del progetto**: ti permette di capire il modo in cui il codice funziona nel suo insieme, e di sapere quali parti di codice verranno affette da una modifica.
- **Documenta l'architettura, non solo il codice**: quando scrivi la documentazione devi essere in grado di tenere in mente la visione d'insieme e aiutare altri sviluppatori a capire come tutte le diverse componenti del codice lavorano insieme. Perciò, quando documenti il tuo progetto concentrati su diagrammi e spiegazioni chiare.
- **Aggiorna la documentazione assieme al codice**: il miglior modo per mantenere aggiornata la documentazione è di modificarla mano a mano che modifichi il codice.
- Se usi uno strumento di gestione del codice sorgente, in ciascun *commit* che modifica il sorgente puoi includere anche le modifiche alla relativa documentazione (vedi in particolare la scheda *"Gestisci il codice sorgente"*).
- **Nella documentazione, non scordare di trattare la parte relativa alla sicurezza**. In particolare, puoi documentare le differenti scelte di configurazione della tua applicazione, e spiegare quali regolazioni sono più sicure.

Controlla la qualità del codice e della documentazione

- Un codice di qualità richiede **l'adozione di buone pratiche e convenzioni di scrittura**, applicate in modo consistente in tutta l'applicazione. È anche opportuno fare riferimento a *convenzioni esistenti*. Ecco alcuni esempi di buone pratiche:
  - **Usare nomi di variabile e di funzione espliciti** rende più semplice capire a colpo d'occhio cosa sta succedendo.
  - **Indentare correttamente il codice** ti permette di comprendere più velocemente la struttura del codice.
  - **Evitare la ridondanza del codice** riduce la necessità di replicare una modifica in più punti. Una svista si dimentica in fretta.
- **Esistono strumenti che possono aiutarti a controllare la qualità del codice**. Una volta che li hai configurati correttamente, ti eviteranno di rileggere il codice per assicurarti che le convenzioni di scrittura siano state rispettate. Alcuni esempi di questi strumenti sono:
- **Ambienti di sviluppo integrati** ("*IDE–Integrated Development Environments*"), magari con l'aggiunta di plugin, che possono essere configurati per rispettare le regole di indentazione, le linee vuote fra diverse porzioni di codice o la posizione delle parentesi, graffe o di altro tipo.
  - **Software di misurazione della qualità del codice** possono segnalarti duplicazioni, mancate adesioni alle regole di programmazione e potenziali bug.

Scheda n°11: Testa le applicazioni

Testare i tuoi prodotti ti permette di controllare che funzionino correttamente, che l'esperienza utente sia buona e di trovare e prevenire difetti prima che il codice vada in produzione. I test riducono anche i rischi di violazioni dei dati personali.

Automatizza i test

- I **test di sviluppo** (unit, funzionali, ecc.) verificheranno la corrispondenza fra le specifiche e il funzionamento del prodotto. I **test di sicurezza** (test con dati casuali anche detti *"fuzzying"*, scansione delle vulnerabilità, ecc.) verificheranno che il prodotto continui a funzionare in modo accettabile anche al di fuori dell'utilizzo normale, e che non presenti vulnerabilità che permetterebbero a terze parti di comprometterne la sicurezza. Entrambi i tipi di test sono importanti per il corretto funzionamento della tua applicazione.
- Appronta un **sistema di integrazione continua** per eseguire i test in automatico dopo ogni modifica al codice sorgente.

Integra i test nella tua strategia di business

- Aggiungi l'implementazione dell'ambiente di test alla strategia di business. Le **metriche accettabili** devono essere definite congiuntamente da tutte le parti in causa prima dello sviluppo..
- Le metriche che puoi considerare sono per esempio:
  - Il **tasso di copertura dei test** e il loro tipo;
  - il **tasso di duplicazione** nel codice;
  - il **numeri di vulnerabilità** (come definite dagli strumenti) e il loro tipo, ecc.

Fai attenzione ai dati di test!

- Non si dovrebbero usare dati "veri" di produzione nelle fasi di sviluppo e di test. Usare dati personali per fare dei test dal tuo database di produzione equivale a **distoglierlo dal suo scopo originario**.
- Se usi dati personali al di fuori della produzione, occorre ricordare che i **rischi di sicurezza aumentano**: accesso ai dati da parte di persone che non ne hanno necessità, molteplici posizioni di archiviazione, ecc.
- Per questo motivi devi costituire un **dataset fittizio** che assomigli ai dati che verranno effettivamente trattati dalla tua applicazione. Un dataset fittizio garantirà che la comunicazione di questi dati non abbia alcun impatto sulle persone.
- Se devi **importare configurazioni esistenti** dalla produzione nei tuoi test, considera l'**anonimizzazione dei dati** presenti.

Scheda n°12: Informa gli utenti

Il principio di trasparenza del GDPR richiede che ogni informazione relativa al trattamento di dati personali sia concisa, trasparente, comprensibile e facilmente accessibile in linguaggio chiaro e semplice.

Chi informare, e quando?

- Gli interessati devono essere informati:
  - sia **nel caso di raccolta diretta dei dati**, per esempio quando i dati sono raccolti direttamente dagli interessati (ad es. moduli, acquisti online, sottoscrizione di un contratto, apertura di un conto corrente bancario) oppure tramite strumenti o tecnologie che monitorano l'attività delle persone (ad es. analisi della navigazione Internet, geolocalizzazione e analytics o tracciamento Wi-Fi per la misura del pubblico, ecc.);
  - che **nel caso di raccolta indiretta di dati personali**, quando i dati non sono raccolti dai diretti interessati (ad es.: dati ottenuti da partner commerciali, *data broker*, fonti pubblicamente disponibili, o altro).

- Le informazioni devono essere fornite:
  - **prima dell'avvio della raccolta di dati**, nel caso di raccolta diretta;
  - **il prima possibile nel caso di raccolta indiretta** (tutt'al più al primo contatto con la persona) e non oltre un mese dalla raccolta (con alcune *eccezioni*);
  - **nel caso di modifiche sostanziali al trattamento o di eventi particolari**. Per esempio: nuova finalità, nuovi destinatari, modifiche al modo in cui si possono esercitare i diritti, *data breach*.

Quali informazioni devo fornire?

- In tutti i casi, devi specificare:
  - **L'identità e i contatti dell'organizzazione** che tratta i dati (chi tratta i dati?) ;
  - **Le finalità** (per cosa saranno usati i dati raccolti?);
  - **La base giuridica** su cui si fonda il trattamento (vedi tutte le ***informazioni sulla base giuridica***);
  - **La natura obbligatoria o facoltativa della raccolta di dati** (la qual cosa richiede una riflessione a monte riguardo all'utilità della raccolta rispetto alla finalità perseguita – il principio di "minimizzazione" dei dati) e le **conseguenze per l'interessato** nel caso decida di non fornire i dati;
  - **I destinatari o le categorie di destinatari dei dati** (chi, per le finalità dichiarate, ha bisogno di accedere ai dati, o di riceverli, inclusi eventuali responsabili esterni?) ;
  - **Il periodo di conservazione dei dati** (o i criteri per determinarlo);
  - **L'esistenza dei diritti dell'interessato e il modo in cui può esercitarli** (diritti di accesso, rettifica, cancellazione, opposizione sono applicabili a tutte le operazioni di trattamento) ;
  - **I contatti del responsabile per la Protezione dei Dati/Data Protection Officer** dell'organizzazione, se nominato, o della persona che segue le problematiche relative alla protezione dati;
  - **Il diritto di presentare reclamo all'Autorità Garante.**

- In certi casi particolare bisogna fornire informazioni ulteriori, come per esempio nel caso che i dati siano trasferiti al di fuori della UE, nel caso di profilazione o decisioni automatizzate, o quando la base giuridica del trattamento è il legittimo interesse di chi raccoglie i dati (Titolare); (v. le *linee guida sulla trasparenza* per ulteriori informazioni).
- Nel caso di raccolta indiretta dei dati, occorre specificare anche:
  - **Le categorie di dati** raccolti;
  - **L'origine dei dati** (indicando in particolare se vengono da fonti pubblicamente disponibili).

In quale forma devo fornire queste informazioni (informativa)?

- Le informazioni devono essere **di facile accesso**: l'utente deve essere in gradi di trovarle senza difficoltà.
- **Devono essere fornite in modo chiaro e comprensibile**, ad es. con un vocabolario ridotto (frasi brevi, niente termini tecnici o legali, niente ambiguità) e con informazioni adattate al tipo di pubblico (con particolare attenzione ai minori e alle persone vulnerabili).
- **Devono essere scritte in modo conciso**. Per evitare che troppe informazioni distolgano l'utente, è necessario **fornire le informazioni più rilevanti al momento giusto**.
- Le informazioni relative alla protezione dei dati devono essere **distinguibili da informazioni non specificamente relative alla privacy (come clausole contrattuali o termini generali e condizioni d'uso).**

Cosa devo comunicare quando la sicurezza dei dati viene compromessa?

- **Un'organizzazione può, per errore o negligenza, accidentalmente o in seguito a un attacco, subire una violazione di dati personali, cioè la perdita, alterazione, distruzione o distribuzione non autorizzata di dati**. In questo caso, se l'evento può rappresentare un rischio per i diritti e le libertà fondamentali degli interessati, l'organizzazione deve comunicare l'evento alla propria Autorità Garante **entro 72 ore**.
- Se questi rischi sono elevati, l'organizzazione deve anche informare il prima possibile gli interessati, fornendo loro indicazioni su come proteggere i propri dati (ad es. cancellazione di carte di credito compromesse, modifica di password, modifica dei parametri di sicurezza, ecc.).
- La notifica della violazione all'Autorità Garante deve avvenire tramite il modulo *sul sito del Garante*

Risorse utili

- Il sito *Data & Design* (in Inglese) sviluppato dal Digital Innovation Laboratory di CNIL sviluppa questi concetti e contiene *esempi di interfaccia*.
- Il sito di CNIL e del Garante contengono anche *molti esempi in Francese* e in italiano.
- La *pagina sulle violazioni dei dati personali* sul sito di CNIL (in Francese)) e del Garante (in Italiano).

Scheda n°13: Preparati all'esercizio dei diritti degli interessati

Le persone di cui tratti i dati mantengono dei diritti su quei dati: diritto di accesso, di rettifica, di opposizione, di cancellazione, di portabilità dei dati e di limitazione del trattamento. Devi fornire loro gli strumenti per esercitare in modo effettivo i loro diritti, e allo stesso tempo includere nei tuoi sistemi informatici tutti gli strumenti informatici necessari perché quei diritti possano essere adeguatamente rispettati.

Definendo in anticipo le modalità con cui potranno contattarti, e i modi con cui risponderai alle loro richieste, sarai in grado di gestire in modo efficace il loro esercizio dei diritti.

Misure minime da attivare

- All organisations that use personal data have **the obligation to indicate where and how** individuals can exercise their rights in relation to this data. For example, you can mention an e-mail address or a web form when informing individuals, as well as in your privacy policy.
- In order to facilitate the exercise of people's rights, these rights may also be **implemented**, in whole or in part, directly in **the application or software you develop**. This specific implementation is not mandatory, but it allows you to meet users' expectations and reduce the time and complexity of processing this type of request.
- Above all, in case of access or operations directly performed by a person who exercises his or her rights, do not forget to manage his **authentication** in a secure way. Overall, **trace** also all operations that have an impact on his or her personal data.

- Ogni organizzazione che usa dati personali ha **l'obbligo di indicare dove e come** le persone interessate possono esercitare i propri diritti relativamente a quei dati. Per esempio, puoi indicare un indirizzo email o un form web al momento in cui informi le persone del trattamento o nella tua privacy policy.
- Per facilitare l'esercizio dei diritti delle persone, questi diritti possono anche essere **implementati**, in tutto o in parte, direttamente **nell'applicazione o nel software che sviluppi**. Questo ti permette di venire incontro alle aspettative degli utenti e di ridurre il tempo e la complessità di trattare le loro richieste.
- Soprattutto, sia nell'accesso ai propri dati che nelle operazioni effettuate direttamente da una persona sui propri dati in tuo possesso, non dimenticare di gestire **l'autenticazione** in modo sicuro. Ad ogni modo, **tieni traccia** di tutte le operazioni che hanno un impatto sui suoi dati personali.

Alcuni esempi di diritti e di una loro possibile implementazione

- **Diritto di accesso**: le persone hanno il diritto di ottenere una copia di tutte le informazioni su di loro di cui sei in possesso. Questo, fra l'altro, permette a una persona di sapere se dati che la riguardano sono oggetto di trattamento e di ottenerne una copia in un formato di uso comune. In particolare, la persona può controllare la correttezza delle informazioni.

Possibile implementazione: Prevedi una funzionalità che consenta la visualizzazione di tutti i dati relativi a uno specifico interessato. Se la quantità di dati è eccessiva, offri all'interessato la possibilità di scaricare un archivio contenente tutti i suoi dati.

- **Diritto di cancellazione**: le persone hanno il diritto di chiedere la cancellazione di tutti i dati di cui disponi su di loro.

Possibili implementazioni:

1. Prevedi una funzionalità che cancelli tutti i dati di una persona.
2. Prevedi anche una notifica automatica a tutti i responsabili esterni perché a loro volta cancellino i dati relativi a quella persona.
3. Prevedi anche la possibilità di cancellare i dati anche dai backup, o fornisci una soluzione alternativa che, quando si ripristina un backup, non ripristini i dati cancellati.

- **Diritto di opposizione**: in alcuni casi le persone hanno il diritto a opporsi al trattamento dei loro dati per uno scopo specifico.
  Possibile implementazione: prevedi una funzionalità che permetta agli interessati di esprimere la propria opposizione al trattamento. Quando l'interessato esercita il proprio diritto di opposizione in questo modo, il Titolare del trattamento deve cancellare i dati già raccolti, e deve astenersi dal raccogliere altri dati relativi alla persona in questione.
- **Diritto alla portabilità dei dati**: le persone hanno il diritto a ricevere i propri dati in un formato leggibile da computer per il proprio uso personale o per il trasferimento a un'altra organizzazione
  Possibile implementazione: Prevedi una funzionalità che consenta alle persone di scaricare i propri dati in un formato machine-readable standard (CSV, XML, JSON, etc.).
- **Diritto di rettifica**: le persone hanno il diritto a chiedere la modifica dei propri dati quando questi siano scorretti, al fine di evitare la disseminazione di informazioni erronee.
  Possibile implementazione: consenti la modifica diretta dei dati nell'account dell'utente.
- **Diritto alla limitazione del trattamento**: le persone hanno il diritto a chiedere la sospensione per un certo periodo di tempo del trattamento dei propri dati, ad esempio per valutare una loro disputa riguardo al trattamento dei loro dati o per dar loro il tempo di esercitare i propri diritti. ***Possibile implementazione***: Permetti agli amministratori di sistema di mettere "in quarantena" i dati di una persona: da quel momento quei dati non possono più essere né letti né modificati.

In conclusione

- Il *sito Data & Design* sviluppato dal Digital Innovation Laboratory di CNIL sviluppa questi concetti e contiene *esempi di interfacce per l'esercizio dei diritti*.
- E poi, lascia andare la tua **inventiva**!(In caso di dubbi, puoi aconsultare la CNIL o il Garante nelle forme previste).

Scheda n°14: Definisci un periodo di conservazione dei dati

I dati personali non possono essere conservati per un periodo di tempo indefinito: il periodo di conservazione deve essere determinato sulla base dello scopo del trattamento. Una volta che lo scopo è stato raggiunto, i dati devono essere archiviati (se esiste un obbligo di legge di conservazione), cancellati o anonimizzati (ad esempio, per produrne delle statistiche).

Cicli di conservazione dei dati

- Il ciclo di conservazione dei dati personali può essere diviso in **tre distinte fasi successive**:
  - il database attivo;
  - l'archiviazione intermedia;
  - l'archiviazione finale o la cancellazione.
- I meccanismi per la cancellazione di dati personali dai database attivi assicurano che i dati siano conservati ed acceduti da parte dei servizi operativi solo **per il tempo necessario al raggiungimento dello scopo del trattamento** .
- Assicurati che **i dati non sono mantenuti in database attivi** semplicemente **catalogandoli come archiviati** . I dati archiviati (archivio intermedio) devono essere accessibili solo a uno specifico servizio responsabile della loro cancellazione dall'archivio se necessario.
- Assicurati di avere **specificato dei modelli di accesso** per i dati archiviati, perché l'accesso a un archivio deve avvenire solo per motivi specifici o eccezionali.
- Se possibile, utilizza la stessa implementazione tanto per **la cancellazione o l'anonimizzazione** quanto per il **diritto alla cancellazione** (vedi [scheda 13 sull'esercizio dei diritti](Scheda_n°13:*Preparati*all'esercizio_dei_diritti_degli_interessati)), in modo da garantire un funzionamento omogeneo del tuo sistema.

Alcuni esempi di tempi di conservazione (validi in Francia)

- I **dati relativi agli stipendi o al computo orario degli impiegati** possono essere conservati per 5 anni.
- I **dati in un archivio medico** devono essere conservati per 20 anni.
- I **dati di un potenziale cliente che non risponde ai solleciti** possono venire conservati per 3 anni.
- I **dati di log** possono essere conservati per 6 mesi.

Scheda n°15: Considera la base giuridica durante l'implementazione tecnica

Il trattamento dei dati personali si deve basare su una delle "basi giuridiche" indicate nell'*Articolo 6 del GDPR*. La base giuridica di un trattamento è in un certo senso la giustificazione dell'esistenza del trattamento. La scelta di una base giuridica ha una ricaduta diretta sulle condizioni per implementare le operazioni di trattamento e i [diritti degli interessati](#Scheda_n°13_:*Preparati*all'esercizio_dei_diritti_degli_interessati). Per questo, individuare le basi giuridiche del trattamento prima di avviare qualsiasi sviluppo ti aiuterà a includere le funzioni necessarie ad assicurare che tutte le operazioni di trattamento rispondano ai requisiti di legge e rispettino i diritti delle persone.

Definizione delle basi giuridiche nel GDPR

- Nell'ambito di un'organizzazione privata, le basi giuridiche usate per sviluppare un trattamento di dati sono di solito:
  - **Contratto**: il trattamento è necessario per la preparazione o la realizzazione di un contratto fra l'interessato e l'entità che effettua le operazioni di trattamento;
  - **Legittimo interesse**: l'organizzazione ha un "legittimo" interesse a eseguire le operazioni di trattamento, purché non vengano meno i diritti e le libertà degli interessati.;
  - **Consenso**: L'interessato ha dato il suo esplicito consenso al trattamento dei propri dati.
- Un ente o una autorità pubblica che svolgono compiti di pubblico interesse possono usare anche altre basi giuridiche:
  - **Obbligo di legge**: il trattamento è richiesto da un testo normativo.
  - **Compito di pubblico interesse**: il trattamento è necessario per svolgere un compito nel pubblico interesse.
- Infine, in casi molto specifici, la base giuridica può essere la **tutela degli interessi vitali**, per esempio in casi di emergenza sanitaria.

Scegli la base giuridica appropriata

- Per prima cosa, controlla sul sito di CNIL o del Garante che non ci siano **norme che impongono specifici vincoli** (per esempio: *cookie e altri tracciatori*).
- **Una sola base giuridica può essere scelta** per una data finalità. Non si possono cumulare più basi giuridiche per una stessa finalità. Le stesse operazioni possono rispondere a più di una finalità, e ciascuna di esse deve avere la propria base giuridica.
- Come detto sopra, nel caso di una **pubblica autorità**, l'obbligo di legge e il pubblico interesse saranno le basi giuridiche più rilevanti nella maggior parte dei casi.
- Se le operazioni di trattamento sono parte di una relazione contrattuale e il loro scopo è oggettivamente e strettamente necessario per la fornitura del servizio all'utente (ad es. nome, cognome e indirizzo per creare un account su un sito di commercio elettronico), allora la base giuridica più appropriata dovrebbe essere il **contratto**.
- Se il trattamento non è parte di una relazione contrattuale con l'utente, allora puoi appellarti **alla base giuridica del consenso o del legittimo interesse**. Se il trattamento è potenzialmente intrusivo (ad esempio, raccolta di dati di geolocalizzazione, ecc.) allora **è probabile che la base giuridica appropriata sia il consenso**.
- Se il trattamento riguarda **dati sensibili** (dati riguardanti la salute, orientamento sessuale, politico, ecc.) allora, oltre alla base giuridica, devi identificare anche un'eccezione *nell'articolo 9 del GDPR* al divieto di trattamento.

Esercizio dei diritti e informazione da fornire a seconda della base giuridica

- La tabella qui sotto riassume i diritti che possono essere esercitati per ciascuna base giuridica:

	Diritto di accesso	Diritto di rettifica	Diritto di cancellazione	Diritto di limitazione	Diritto alla portabilità dei dati	Diritto di obiezione
Consenso	✓	✓	✓	✓	✓	**Ritiro del consenso**
Contratto	✓	✓	✓	✓	✓	✗
Legittimo interesse	✓	✓	✓	✓	✗	✓
Obbligo di legge	✓	✓	✗	✓	✗	✗
Pubblico interesse	✓	✓	✗	✓	✗	✓
Tutela degli interessi vitali	✓	✓	✓	✓	✗	✗

- La base giuridica deve sempre comparire nell'informativa fornita alla persona interessata.
- **Se il trattamento si basa sul legittimo interesse**, devi anche indicare quale interesse intendi perseguire (contrasto alle frodi, sicurezza del sistema, ecc.).
- Ti raccomandiamo di **documentare la tua scelta di una base giuridica**. Per esempio, puoi riportare queste scelte in una mappa dei processi o includerle nella documentazione tecnica.

Il caso specifico dei cookie e degli altri tracciatori

- La Direttiva Europea ePrivacy richiede il consenso dell'utente prima di qualsiasi azione che registri informazioni (tramite cookie, identificatori o altri tracciatori come fingerprint e pixel) o che acceda a informazioni registrate nel dispositivo dell'utente.
- Viene fatta un'eccezione quando i cookie hanno il solo scopo di consentire comunicazioni elettroniche, o sono strettamente necessari per fornire un servizio richiesto dall'utente.
- Inoltre, l'uso di un singolo tracciatore per una pluralità di scopi non ti esenta dall'ottenere il consenso per ciascuno degli scopi che lo richiedono. Per esempio, se un cookie di autenticazione viene usato anche per consentire pubblicità mirate, devi ottenere il consenso per questo secondo scopo, come faresti se l'utente non fosse autenticato.

Scheda n°16: Usa le analytics nei tuoi siti e applicazioni

Gli strumenti per la misurazione del pubblico sono utilizzati per ottenere informazioni riguardo alla navigazione dei visitatori di un sito o di una applicazione mobile. In particolare, rendono possibile comprendere in che modo gli utenti arrivano al sito e di ricostruire il loro percorso al suo interno. I cookie sono soggetti a consenso, eccetto in un caso particolare. Nota che questa sezione fa riferimento alla direttiva ePrivacy e può essere soggetta a variazioni su base nazionale. Informati presso la tua Autorità Garante per la Protezione dei Dati Personali per conoscere la sua posizione al riguardo.

Ottenere il consenso

- In generale, **prima di caricare o leggere un cookie o un tracciatore**, chi produce un sito o un'applicazione deve:
  - informare gli utenti Internet riguardo allo scopo dei cookie;
  - ottenere il loro consenso;
  - fornirgli la possibilità di rifiutarlo.
- A meno di ricadere nel perimetro esatto descritto nel prossimo paragrafo, **questo obbligo si applica ai tracciatori usati per la misurazione del pubblico sui siti web**.

Per beneficiare dell'esenzione dal consenso

- **A seconda di un certo numero di condizioni**, i cookie usati per la misurazione del pubblico sono esentati dal consenso.
- **Queste condizioni, specificate nelle *linee guida per i cookie e altri tracciatori (in Inglese)*linee guida per i cookie e altri tracciatori (in Inglese), per la Francia e sul sito del Garante per l'Italia, sono**:
- Informare gli utenti del loro uso;
- Dare loro la possibilità di non acconsentire al loro uso;
- Limitarsi ai soli scopi che seguono:
  - misurazione del pubblico;
  - A/B testing;
- Non incrociare i dati trattati con altri trattamenti (schede cliente, statistiche sulle visite ad altri siti, ecc.);
- Limitare lo scopo del tracciatore a un singolo sito o applicazione;
- troncare l'ultimo bit dell'indirizzo IP;
- Limitare la vita del tracciatore a 13 mesi.
- Se queste condizioni sono soddisfatte, possiamo **passare da un regime opt-in (soggetto a consenso) a un regime opt-out**.
- È anche possibile che una singola terza parte (fornitore) fornisca un servizio di misura comparativa del pubblico per più siti o applicazioni, a patto che i dati siano raccolti, trattati e registrati indipendentemente per ciascun sito o applicazione e che i tracciatori siano indipendenti l'uno dall'altro.

In pratica

- **La maggior parte delle offerte riguardanti la misura del pubblico non ricadono nello scopo dell'esenzione, indipendentemente da come vengono configurate**.

- Per beneficiare di questa esenzione, contatta il tuo solution provider, o usa software open source come Matomo che puoi configurare autonomamente.