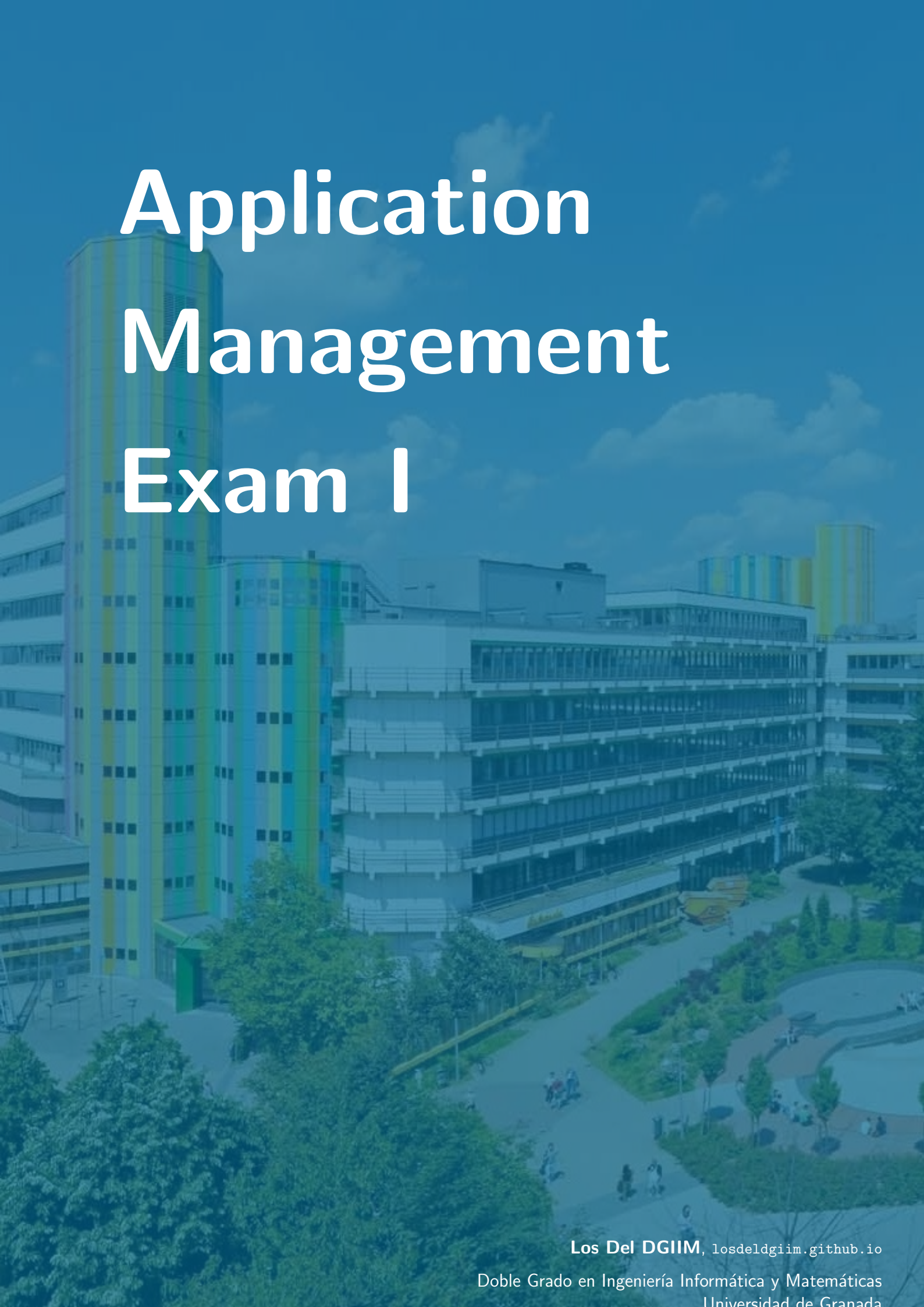
Application Management Exam I



Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas

Universidad de Granada

Application Management Exam I

Los Del DGIIM, `losdeldgiim.github.io`

Arturo Olivares Martos

Granada, 2025-2026

Asignatura Application Management.

Curso Académico Winter Semester 2024-25.

Fecha 13 de Febrero de 2025.

Ejercicio 1 (Application Lifecycle Management (ALM))**.**

1. (1 Punkt) Nennen Sie vier Phasen des ALM, welche auch dem Wasserfallmodell zugeordnet werden können.
2. (2 Punkte) Welche(n) Vorteil(e) bietet die Wahl einer permissiven Lizenz wie Apache 2.0 oder MIT?
   a) Förderung der breiten Nutzung und Akzeptanz.
   b) Strikte Kontrolle über alle Modifikationen.
   c) Geringere Hürden für kommerzielle Nutzung.
   d) Verpflichtung zur Offenlegung aller Änderungen.
3. (2 Punkte) Welche der folgenden Aussagen zum Application-Lifecycle-Management (ALM) trifft/treffen zu?
   a) ALM umfasst neben Entwicklung auch Betriebs- und Wartungsprozesse.
   b) ALM ist nur relevant für agile Softwareprojekte.
   c) ALM setzt sich häufig aus kontinuierlichem Monitoring und Feedback zusammen.
   d) Ein zentrales Ziel von ALM ist die Transparenz über die gesamte Lebensdauer einer Software.
4. (2 Punkte) Welche der folgenden Aussagen trifft/treffen auf die Shift-Left-Teststrategie zu?
   a) Tests möglichst spät durchführen, damit mehr Zeit für die Entwicklung bleibt.
   b) Probleme möglichst früh erkennen und beheben, um Kosten zu senken.
   c) Den Testaufwand im Betrieb auf ein Minimum reduzieren.
   d) Nur kritische Funktionen in einer frühen Phase testen.
5. (2 Punkte) Warum ist es wichtig, den gesamten Lebenszyklus einer Software zu betrachten?
6. (2 Punkte) Was versteht man unter 'Flakiness' bei Tests?
7. (2 Punkte) Beschreiben Sie, welche Einfluss 'flaky' Tests auf eine CI/CD haben.
8. (4 Punkte) Nennen Sie vier Stakeholder, welche typischerweise in den ALM-Prozess involviert sind und beschreiben Sie diese kurz.

Ejercicio 2 (Versionskontrolle)**.**

1. (2 Punkte) Welche der folgenden Vorteile bietet ein verteiltes Versionskontrollsystem?
   a) Jeder Entwickler hat eine vollständige Kopie des Repositories.

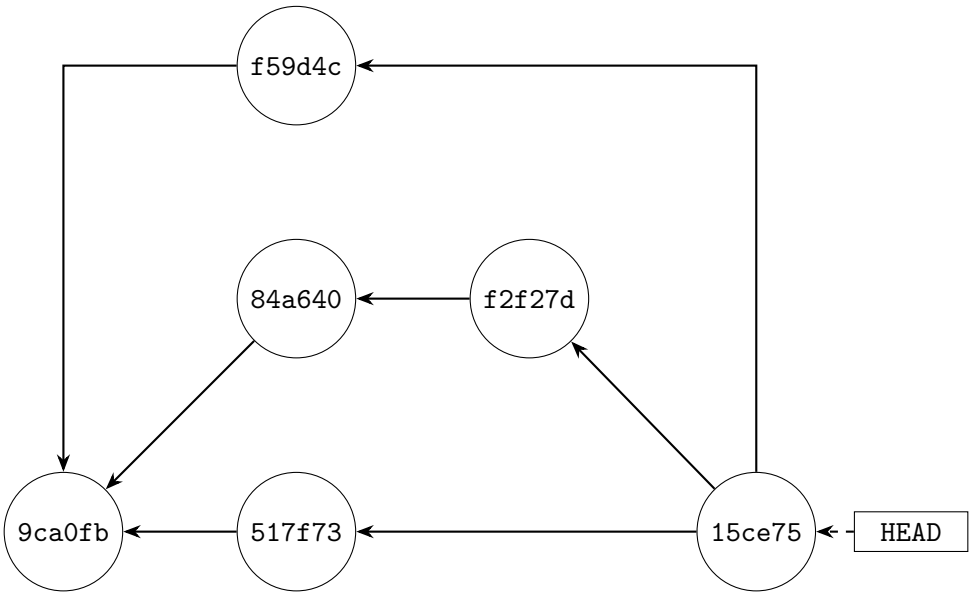


Figura 1: Commit-Historie

b) Änderungen werden automatisch mit allen anderen synchronisiert.

c) Es gibt immer einen zentralen Server, der alle Änderungen speichert.

d) Arbeiten an Branches können unabhängig voneinander erfolgen.

2. (2 Punkte) Welche Rolle spielen Hashes bei Git?

   a) Sie identifizieren eindeutig jede Version einer Datei oder eines Commits.

   b) Sie verhindern Datenverlust bei Netzwerkausfällen.

   c) Sie ermöglichen es Entwicklern, Konflikte automatisch zu lösen.

   d) Sie gewährleisten die Integrität der gespeicherten Daten.

3. (6 Punkte) Diskutieren Sie die Vor- und Nachteile von Rebase vs. Merge in Bezug auf Historie, Konfliktmanagement und Teamabsprachen aus der Sicht:

   - Der Übersichtlichkeit der Historie
   - Des Konfliktmanagements
   - Der Teamabsprachen (z. B. wann force push nötig wird).

4. (5 Punkte) Es wurde folgender Befehl ausgeführt: `git checkout HEAD^2`. Auf welchen Commit zeigt `HEAD`? Begründen Sie Ihre Antwort. In Figur 1 ist die Commit-Historie dargestellt, und in Listing 1 die Historie von `HEAD`.

5. (4 Punkte) Ein Entwicklerteam verwendet Git für ein Webprojekt. Aus Versehen hat ein Teammitglied in einem früheren Commit eine Datei `.env` eingecheckt, in der sensible Zugangsdaten (API-Keys, Datenbankpasswörter) enthalten sind. Das ist dem Team erst nach einigen Tagen aufgefallen, nachdem bereits weitere Commits und Branches erstellt wurden. Das Repository ist öffentlich auf GitHub verfügbar. Die Git-Dokumentation beschreibt den Befehl `git revert` wie in Listing 2 dargestellt. Begründen Sie, wie und ob dieser

```
HEAD Historie:
15ce75 merge f2f27d, f59d4c
517f73 checkout
f2f27d commit
84a640 checkout
f59d4c commit
9ca0fb checkout
```

Código fuente 1: `HEAD` Historie

```
1. git revert [--no-edit] [-n] [-m <parent-number>] [-s] [-S[keyid]] <commit>...
2. git revert [--continue | --skip | --abort | --quit]

> Wenn ein oder mehrere bestehende Commits angegeben werden, werden die durch
die zugehörigen Patches eingeführten Änderungen rückgängig gemacht. Dabei
entstehen neue Commits, die diesen Vorgang festhalten. Voraussetzung dafür ist,
dass dein Working Tree sauber ist.
```

Código fuente 2: Git Revert Befehl

Befehl dazu beitragen kann, das Problem zu beheben. Sollten der Befehl nützlich sein, geben Sie außerdem den abgeätzten Befehl an, welcher das Problem behebt.

6. (2 Punkte) Reicht das Entfernen aus Git aus, um die Sicherheit vollständig wiederherzustellen? Begründen Sie Ihre Antwort.

7. (2 Punkte) Welche Maßnahmen würden Sie dem Team vorschlagen, um solche Vorfälle in Zukunft zu vermeiden? Nennen Sie mindestens zwei.

Ejercicio 3 (Deployment und Delivery)**.**

1. (2 Punkte) Welche der folgenden Aussagen trifft/treffen auf Continuous Deployment zu?

   a) Jede Codeänderung durchläuft manuelle Tests, bevor sie deployed wird.

   b) Jede erfolgreich getestete Codeänderung wird automatisch in Produktion deployed.

   c) Continuous Deployment erfolgt nur bei erfolgreichen Software-Releases.

   d) Continuous Deployment setzt keine vollständige Automatisierung voraus.

2. (2 Punkte) Welche der folgenden Aussagen zu Canary Releases ist korrekt?

   a) Alle Benutzer erhalten sofort die neue Version.

   b) Die neue Version wird erst einer kleinen Nutzergruppe zur Verfügung gestellt.

   c) Ein Rollback ist nicht möglich.

   d) Canary Releases sind nicht für sicherheitskritische Systeme geeignet.

```yaml
services:
    magicweb1:
        image: MagicWeb
        ports:
            - "8081:8080"
        container_name: magicweb_container1
    magicweb2:
        image: MagicWeb
        ports:
            - "8082:8080"
        container_name: magicweb_container2
```

Código fuente 3: Docker-Compose Datei

3. (3 Punkte) Welche Herausforderungen müssen Unternehmen bei der Umstellung auf automatisiertes Deployment berücksichtigen?

4. (3 Punkte) Continuous Integration, Continuous Delivery und Continuous Deployment sind zentrale Prinzipien moderner Softwareentwicklung. Erklären Sie die Unterschiede dieser Konzepte.

5. (2 Punkte) Nennen Sie zwei Vorteile und zwei Herausforderungen von CI/CD Pipelines

6. (6 Punkte) Sie haben eine Binary (MagicWeb), welche einen Webserver auf Port 8080 startet. Diese Binary haben Sie frisch erstellt. Da diese Binary Ihre kompletten Geheimnisse enthält, möchten Sie nicht, dass die Binary im Internet verfügbar ist, daher haben Sie dafür gesorgt, dass Ihr Rechner nicht mit dem Internet verbunden ist. Sie möchten diese Binary aber zweimal ausführen, deshalb haben Sie Docker gerade frisch installiert. Dazu wollen Sie die Docker-Compose Datei wie in Listing 3 nutzen.

   Wenn Sie nun `docker compose up` ausführen, beschreiben Sie was nun passiert. Sollte der Befehl nicht funktionieren, beschreiben Sie, welche Schritte man vornehmen müsste, um das Problem zu beheben.

Ejercicio 4 (Secure Software Development)**.**

1. (2 Punkte) Welche der folgenden Aussagen trifft/treffen auf Netflix Chaos Monkey zu?

   a) Es simuliert und testet zufällige Ausfälle von Systemen, um die Infrastruktur zu verbessern.

   b) Es dient dazu, sensible Daten in Cloud-Umgebungen zu verschlüsseln und vor Bedrohungen zu schützen.

   c) Es ist ein Tool zur automatisierten Schwachstellenanalyse in Webanwendungen.

   d) Es analysiert den Softwarecode auf unsichere Abhängigkeiten und veraltete Bibliotheken.

2. (2 Punkte) Welche der folgenden Aussagen trifft/treffen zu?

   a) Mutational Fuzzing ist eine Technik, bei der Eingaben durch gezielte synthetische Constraints generiert werden.

   b) Symbolic Execution eignet sich besonders für Programme mit vielen Verzweigungen.

   c) Coverage-Guided Fuzzing nutzt Feedback über erreichte Code-Pfade, um Eingaben systematisch zu generieren.

   d) Symbolic Execution kann auch auf Binärprogramme angewendet werden, ohne zusätzliche Modellierung.

3. (2 Punkte) Warum ist Fuzzing ein effektives Mittel zur Identifizierung von Sicherheitslücken?

4. (3 Punkte) Beschreiben Sie, wie Symbolische Ausführung (SymExe) mit einem SMT-Solver wie Z3 systematisch Pfade in einem Programm analysiert.

5. (2 Punkte) Falls die symbolische Ausführung einen kritischen Pfad entdeckt, welche Informationen würde Z3 dazu typischerweise liefern?

6. (4 Punkte) Nennen Sie zwei Security-Testing-Methoden und geben Sie für jede ein Beispiel.

7. (4 Punkte) Wie verbessert eine mehrstufige Zertifikatskette die Sicherheit?

8. Was ist binary provenance und was kann passieren wenn dies nicht gewährt ist? Nennen Sie zwei Fälle.